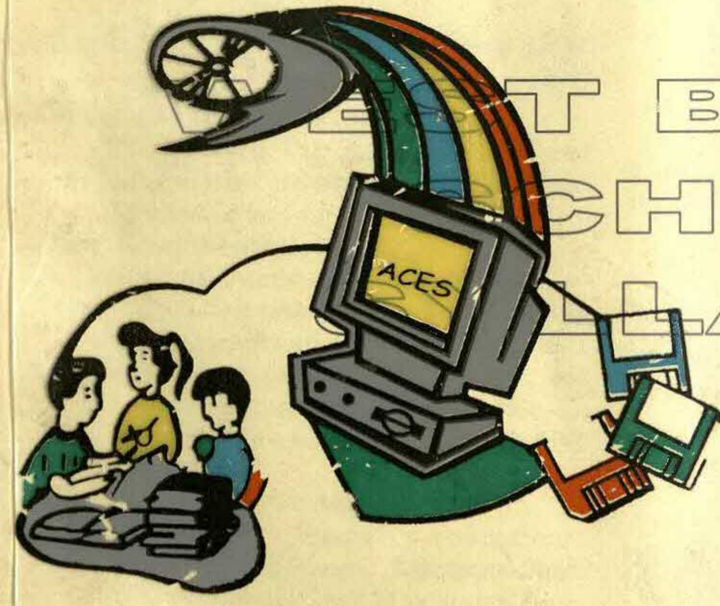
ACES INFOTECH

Gift

The Complete Computer Solution

27



WEST BENGAL SCHOOL SYLLABUS

- ✓ SCHOOL COMPUTER EDUCATION
- ✓ CORPORATE IT TRAINING
- ✓ HARDWARE MAINTENANCE
- ✓ SOFTWARE DEVELOPMENT

ACES INFOTECH PRIVATE LIMITED

An ISO 9001:2000 Certified Company

Corporate Office : ACES HOUSE, 11C, Dover Lane, Kolkata 700 029, India • Phone : (033) 24618110/6370/3715/9202
Website : www.acesinfotech.org • E-mail : aces@cal.vsnl.net.in • Fax : 91-33-246137[illegible]7

HRD Office : 71-B, Alipore Road, Kolkata-700 027 • Phone : (033) 2479982[illegible] 122, 2409[illegible]168, 244[illegible] 19
Fax : 91-33-24794630 E mail :acesinfo198[illegible]@yahoo.com

A & N ISLANDS • ASSAM • ARUNACHAL PRADESH • BIHAR • CHHATTISGARH • DADRA & NAGAR HAVELI
DAMAN & DIU • DELHI • GOA • GUJARAT • JHARKHAND • MADHYA PRADESH • MAHARASHTRA • MANIPUR
MIZORAM • ORISSA • PUNJAB • RAJASTHAN • SIKKIM • TRIPURA • UTTAR PRADESH • WEST BENGAL

WEST BENGAL SCHOOL SYLLABUS

COMPUTER SCIENCE

CLASS-XI

THEORETICAL
Full Marks—70

A. Brief Review of Computer Systems [30M/20P]

Computer Organisation

CPU, Memory, I/O, Storage Devices and other Peripherals. Storage Media, Access Characteristics, etc.

Data Representation

Review of Number Systems-Binary, Octal, Hexadecimal, Conversion of binary to decimals, binary to Octal, Hexadecimal and vice-versa, Different methods of negative number representation-1's complement, 2's complement and signed magnitude.

Topics on Boolean Algebra

Review of logical operations; Combinationals—Logic functions-Basic Gates and realization of complex functions, De-Morgan's theorems, Universal Gates, Multiplexers, Decoder, Encoder, De-Multiplexers.

Boolean Arithmetic

Half Adder, Full Adder, Half Subtractor, Full Subtractor, Multiplication, Booth's Algorithm.

B. Operating systems [10M/10P]

Functions and role of operating system

Familiarities with different commands and utilities; DOS and Windows.

Study of Unix/Linux with respect to commands and utilities.

C. Programming with C [30M/30P]

Concept of Algorithms and Data Structure

Character set, Constants, Variables, Operators, Head section, control structure, loop structure, Arrays, concept of pointers, functions, library functions, structure, concept of files.

Input / Output operations.

Simple problem Solving.

PRACTICAL
Full Marks - 30

A. Familiarization with Computer System and Operating System Commands. [10M/10P]

B. Problem Solving Using C Algorithm Design, Coding, Compilation/Linking/Loading (Solving 10/12 Problems) [20M/20P]

CLASS-XII

THEORETICAL
Full Marks—70

A. Fixed and Floating Point Representation of Real Number, Bit Map representation [5M/5P]

B. Sequential Logic Circuits-Flip-Flops, Registers and Counters-Synchronous and Asynchronous Concepts [5M/5P]

C. Manipulation of Data Structure and I/O Files using C-Language [10M/10P]

D. Computer Networking [25M/25P]

Concept of LAN and WAN, Protocols, TCP/IP.

Concept of Internet and Intranet, IP Address, URL.

Dial-up and Leas
width.
Internet applicati
Telnet, FTP e
Introduction to C
Network Security

E. Introduction to

Record Structure
Access Method
Concept of keys f
Physical Storage c
Basic Concepts of
Normalization (up

Dial-up and Leased Line Connection, Bandwidth.

Internet applications E-mail, Web-browsing, Telnet, FTP etc.

Introduction to Client/Server Computation.

Network Security Concepts.

E. Introduction to DBMS (Relational) [25M/25P]

Record Structure, File Organization and Access Methods.

Concept of keys for data retrieval.

Physical Storage organization.

Basic Concepts of Relational databases.

Normalization (up to 3NF).

Introduction to SQL (exposure to DDL, DML & DCL).

Query Processing and Report generation.

PRACTICAL

Full Marks - 30

Programming for Manipulating Data structures Files [10M/10P]

Interaction with RDBMS and SQL [5M/5P]

Web Interaction [5M/5P]

Project Work and Viva-Voce [10M/15P]

[M/P]= NUMBER OF MARKS/NUMBER OF PERIODS

COMPUTER APPLICATION

CLASS XI

Full Marks : 100

Theoretical
Full Marks : 70

Brief Review of Computer Hardware [10M/10P]

History of Computer, Computer Generations, Block Diagram of Computer System, Input Devices, Output Devices, CPU, Primary & Secondary Memory Systems, CD-ROM, Multimedia Systems

Data Representation [7M/7P]

Review of Number Systems- Binary, Octal, Hexadecimal, Conversion of Binary to decimal, Binary to Octal, Binary to Hexadecimal & Vice-Versa

Topics on Boolean Algebra [7M/7P]

AND, OR, NOT Logic Functions, De Morgan's Theorems, Realization of complex Boolean functions, Universal gates.

Concepts of Computer Software & Languages [15M/15P]

* Importance of Software, System Software Vs Application Software.
* Operating System Overview (Dos & Windows)
* Use of Utility programs-Editor, Compiler, Interpreter
* Programming Languages- Generation of languages, Languages used for problem solving- Scientific, Commercial, Data Manipulation, etc,
  Concept of visual language

Data Processing [16M/15P]

Information & Data, Basic Data types

Packages [15M/15P]

Introduction to Word Processing- Invoking MS Word, Create, Edit & Save a document, Cut & Paste- perform operations on blocks of text, headers & footers, Mail merge, Art work, Clipart & drawing tool use. Document Printing, Printer Setup[10M/10P]

Introduction to Power Point- Creating new presentation, Use of Wizards Different fonts & styles, Inserting pictures drawings, saving on Secondary Storage, Slide show, printing etc. [5M/5P]

Practical Paper
Full Marks : 30

Familiarization with Computer system and Peripherals [7M/7P]

Media-Floppy Disk CDROM, Study of "My Computer" details, Edit, Copy, Delete Files, Creation of Directories, Running Programs

Packages, Studying Windows [7M/7P]

Features, Use of Mouse Buttons etc
Windows Explorer

Use of Words [10M/10P]

Creating Textual documents, drawing Cut & Paste, Erasing, Copy from different Document, Saving, Mail Merge, Printing

Study of Power Point Package [6M/6P]

T

Basic Concepts

Simple B
methods of ne
1's compleme
magnitude re
Character and

Logic Function

Basic Gat
Combination

Data Processi

Record S
Directories/F

Computer Net

Brief in
importance a
Structure- di
Network
Internet & I
Connections.
Internet
Web page de

Data Base Co

Introduct
Base, Table

CLASS XII

Full Marks : 100

Theoretical Paper
Full Marks : 70

Basic Concepts [5M/5P]

Simple Binary Arithmetic, Different methods of negative number representation- 1's complement, 2's complement & signed magnitude representation, ASCII codes for Character and Text representation

Logic Functions [5M/5P]

Basic Gates and realization of simple Combinational functions using gates

Data Processing [5M/5P]

Record Structure, Files Organization, Directories/Folders

Computer Networking [20M/20P]

Brief introduction of Networking importance and application areas, Network Structure- different layers;

Networking features of LAN, WAN, Internet & Intranet; Dial-up, and leased line Connections, ISP, URL;

Internet Services- E-mail, Web browsing, Web page design using HTML

Data Base Concepts [20M/20P]

Introduction to Access, Concept of Data Base, Table creation & manipulation of data using access facilities, addition of field(s) in a Table, inserting, deleting and updating data, Report Generation

Spreadsheet [15M/15P]

Introduction to Excel, Concepts of Worksheet, Entering data in a Worksheet, Entering labels, values & formulas, Saving a Worksheet, Making charts & graphs, perform calculations and recalculations

Practical Paper
Full Marks : 30

Project Work & Study of Email- [15M/30P]

Project Work

Study of system related to School/town/ village/local industries/peoples

i) Using Access creation of Database & populating Tables, Data Manipulation and Report generation

OR

Using Excel- Creation of Marks-Sheets, Balance sheets, Monthly/ yearly expenditures reports etc.

ii) Web page design & implementation

Study of Email

Email send/receive, Use of Search Engines, Collecting information from WEB

Viva-Voce & Sessional [15M]

[M/P]= NUMBER OF MARKS/NUMBER OF PERIODS

Syllabus and curriculum of the additional subject 'Computer Applications'

CLASS – IX

1. History of Computers - 2 periods (Theoretical.)

Origins of Calculation, Abacus, Napier's bone, Babbage's Analytical Engine. Computer Generations : The First Generation, the Second Generation, the Third Generation, the Fourth Generation. Microcomputer, Minicomputer, Main-Frame Computer.

2. Binary System - 6 periods (Theoretical)

The Binary concept, the Binary System, the Binary Arithmetics :

i) Conversion from Decimal to Binary System and vice versa with respect to Integers only.

ii) Negative integer Representation.

iii) Binary Addition, Subtraction with complement, the octal System, the Hexadecimal System, Bit, Byte & Words.

iv) ASCII code and its representation.

3. Elementary Boolean Algebra - 3 periods (Theoretical)

A^{1} or A = not A or A complement

$A+A^{1}=1,\ A.A^{1}=O, A+A=A, A+B=B+A$

$A+BC=(A+B)(A+C)$,

where + means OR (∨) and . means AND (∧).

4. Structure of Computers - 3 periods (Theoretical)

i. Input Devices	Console, Keyboards, Bar Codes Magnetic Ink, Character Reader, touch Screen, Light pen, Mouse, Speech input, Graphic Tablet
ii. General Processing unit	Control Unit, Arithmetic Logic Unit, Memory.
iii. output Devices	Visual Display Unit, Line Printer, Dot Matrix Printers, Laser Printer
iv. Mass storage Unit	Magnetic Tape, Magnetic Disk. Winchester Disk, Floppy Disk. Tape Cartridge.

5. Repesentation of data Information concepts of data Processing ---- 3 periods (Theoretical)

1. Definition of Information and data
2. Basic data types
3. Storage of data
4. Concepts of records, files and directories.

6. Introcution to windows ----- 3 periods (Theoretical) + 3 Periods (Practical)

i. What is an operating system and basics of windows

ii. The User-interface :

a) Using Mouse and Moving Icons on the sereen

b) The my computer Icon

c) The Recycle Bin

d) Status Bar, Start and Menu & Menu-selection

e) Running an Application.

f) Windows Explorer, Viewing of Files, Folders and Directories.

g) Creating and Renaming of files and folders.

h) Opening and closing of different Windows.

7. Basic DOS commands --- 3 periods (Theoretical) + 2 periods (Practical)

i) Comparison of DOS and Windows

ii) Switching between DOS and Windows

iii) Basic DOS Commands :

a) files/ Directory Manipulations

b) Copying of files on Disks

c) Delete/ Undelete of Files

d) Formating a floppy

8. Elements of Word processing --5 periods (Theoretical) + 5 periods (practical)

i) Word Processing Basic :

a) An Introduction to Word Processing

b) Opening Word Processing Package

c) The Menu Bar

d) Using the Help

e) Using the Icons below menu bar

ii) Opening documents and closing documents :

a) Opening documents

b) Save and Save AS

c) Page Set up

d) Printing of documents

e) Display/Hiding of Paragraph Marks and Inter Word Space.

iii) Moving Around in a Document :

a) Scrolling the Document

b) Scrolling by line/paragraph

c) First Scrolling and moving pages

iv) Using a document Help Wizard

retical)

v) Text creation and Manipulation :

- a) Paragraph and Tab Setting
- b) Text Selection
- c) Cut. Copy and paste
- d) Font and size Selection
- e) Bold, Italic and Underline
- f) Alignment of Text, Centre, Left, right and Justify.

vi) Formating the Text :

- a) Changing font, Size and colour
- b) Bullets and Numbering
- c) changing case

vii) Handling Multiple documents :

- a) opening and closing of Multiple Documents
- b) cut, copy and paste across the documents.
- c) Saving of clip Boards.

ctories.

viii) Printing :

- a) Printing
- b) Print preview
- c) Print a selected page

9. Software and languages ----- 4 periods (Theoretical)

- i) Hardware and software
- ii) Operating System
- iii) Utility Programme
- iv) Application packages and programmes.
- v) Computer languages.

Generation of languages and uses and application of languages like BASIC, FORTRAN, COBOL, PASCAL and C.

10. Flow Charts ---------- --- 3 periods (Theoretical)

Flow Charts – symbols, their meanings and uses, Writing Some elementory flow charts.

11. Programme in BASIC (Preliminary topics) - - - 5 Period (Theoretical) + 10 periods (Practical)

i) Constants and Variables :

pace.

- a) The Character Set
- b) Constants
- c) Variables
- d) Naming the Variable
- e) Using LET, PRINT, INPUT, DATA, GOTO, READ, INPUT, LPRINT, IF-THEN, IF-THEN-GOTO, and related flow charts.

ii) Expression in BASIC :

a) Arithmetic Expressions
b) Hierarchy of operations
c) Rules of Arithmatic operators
d) Evaluation of expressions

iii) Print Controls :

a) The Comma control
b) The Semicolon Control
c) The TAB function

CLASS - X

1. Programming in BASIC Advanced topic 16 periods (Theoretical)
+ 9 periods (Practical)

i) Revision of Work done in class IX,

ii) System command in BASIC

a) Refinements in Programming
b) Logical Expressions
c) Library Functions
d) User-defined Functions
e) IF-THEN-ELSE, ON-GOTO

iii)

a) Looping - The For-Next statement
b) A subroutine Limited to use of GOSUB - RETURN
c) Arrays (one and two dimentional)
(Single and double-subscripted Variables)

iv) Graphic Features.

2. Data processing - - - - - 4 periods (Theoretical)

i) File based and DBMS based data processing.
File creation
File Access
File Manipulation and maintenance
File in BASIC (Sequential only)
Concept of data based language

3. Project work - - - - - - - 10 periods

Candidate will be equired to submit two reports, each of about 8 pages including diagrams, as follows :

i) A short report on the design and preparation of a programme for a user, This should include :

a) The user's Specification.
b) A restatement of the problem from the programmer's point of view and the programme specifications :
c) Description of the algorithm
d) The programme (suitably annotated)
e) Evaluation of Computer results

ii) A short reporting on the testing of a file-handling programme (using Sequential files in BASIC) this report should include :

a) Specification of the programme design
b) Details of relevant files.

Suggested Application areas

i) Computer Aided Learning/ Instructions
ii) Computers in Bank
iii) Computers in Household application
iv) Computers in Commercial Organisations
v) Computers in Industry
vi) Using laboratory date to varify a law or principle in Science, e.g., Newton's laws of motion, cooling curves for various substances OR other suitable projects.

5. Either I. Computer communication and Internet

or

II. Uses and application of MS office package

5 periods (Theoretical) +5 periods (Prac.)

I. Computer Communication and Internet

i) a) Basic of Computer networks

LAN
WAN

b) Internet

Concept of Internet.
Application of Internet

c) Service of Internet.

www and web-sites
Electronic mails
Communication on Internet.

ii) www and web Browsers

a) web Browsing software

Internet Explorer
Netscape Communicator

b) Surfing the Internet

giving the URL address
Search
Moving around in a web-site
Printing of saving portion of web pages
Down loading

ws :

c) Charting on Internet.

iii) E-mail

a) Basic of electronic mail

What is an eletronic mail
E-mail addressing
Mail box : Inbox and Outbox

b) Using E-mails

Viewing an E-mail
Sending an E-mail
Saving mails
Sending same mail to various users

c) Document handing

Sending Gift copy as attachment
Enclosures to email
Sending a portion of document as email

or

II. Uses & Application of MS Office Package

a) Introduction to Access ...

Concept of database creation & manipulation using Access, Table creation addition of field data entry, data manipulation, report generation.

b) Introduction of Excel ...

Concept of spread sheet, programming spread sheet, inserting data & formula in cells, populating data both from database and from keyboard.

c) i) Use of Access for creating database for your school/village.

ii) Use of Excel for creating marksheet. Bills etc.

EQUIPMENT NECESSARY (with proper arrangements for maintenance)

1. Pentium/Celeron 32 bit with 64 MB RAM, 8.4 GB HDD, 1.44 MB, Floppy, CD Drive. Mouse, Key Board, 14" SVGA colour Monitor, with preloaded windows 95, 98 & NT, MS Office (This should also include Q basic) ...3 Nos.
2. Dot matrix printer 80 Columns.
   or
   Laser Printer.
3. Those offering 5 (I) must have internet facility.

QUALIFICATIONS OF TEACHER :

a) B. Tech. or equivalent/M. Tech. or equivalent in i) Computer Science or ii) Computer. Engineering or iii) Computer Science and Engineering or iv) Computer Science and Technology.

b) B.Sc. (Hons.)/M.Sc.in Computer Science or M.C.A. From a recognized university/Institution.

c) B.Sc. (Hons.) in Physics/Mathematics with i) Computer Science a Pass subject or ii) P.G Diploma in Computer Science from a recognized University/Institution.

d) M.Sc. in physics/Mathematics with P.G Diploma in Computer Science from a recognized University/Institution.

e) Ph. D. in Computer Science from a recognized University/Institution.

f) BCA from recognized University/Institution.

g) B.Sc (Pure) with a diploma in Computer recognized by Government.

MARKS DISTRIBUTION

Full Marks ... 100

A. Theoritical....70 Marks

Topic Title	Marks allotted
1. History of Computer	2
2. Binary State	6
3. Elementary Boolean Algebra	3
4. Structure of Computer	3
5. Representation of data/information concepts of data processing	3
6. Introduction of Windows	3
7. Basic Dos Commands	3
8. Elements of Word processing	5
9. Software and Languages	3
10. Flow charts	2
11. Programme in Basic (Preliminary Topic)	12
12. Programme in Basic (Advanced Topic)	16
13. Data Processing	4
14. Either Communication and Internet or Use and application of MS office Package	5

B. Practical ....20 Marks

Topic Title	Marks Allotted.
1. Intorduction to Windows	2
2. Basic DOS commands	2
3. Elements of Word Processing	3
4. Programming in BASIC (preliminary topic)	5
5. Programming in BASIC (Advanced topic)	5
6. Either Computer Communication and Inernet or Uses and application of MS office pakage	3

C. Project Work — 10 Marks

PRESCRIBED READINGS IN ENGLISH MEDIUM

1. Digital Computer Fundamentals, 6th Edition, T.C. Bartee Mc Graw Hill Kogakusha, Int. Student Ed.
2. Introduction to Computers–Norton–Mc Graw Hill
3. Programming with BASIC–Gottfried–Mc Graw Hill.
4. Manuals for DOS, Windows–95, MS-Office
5. Complete Guide'to Windows 95–Norton, Prentice Hall
6. Bott–Using Microsoft Office-97–Prentice Hall

In Bengali medium the book be publised by West Bengal Board of Secondary Education.

Syllabus of Computer Literacy Training Program (CLTP) in West Bengal

Class VI

1 Introduction Need for Computer Literacy – Speed, Intelligence and merits; History(both analog and digital); early calculating devices- abacus, Napier's bones, difference engine, analytical engine, slide rule, punch card system, tabulator, sorter. [5L]

2 Electronic computer different generations – super, mini, desktop PC, laptop [5L]

3 Basic Components: External view of a typical computer (PC): Keyboard, mouse, monitor, hard disk, floppy, CD, Speaker, modem, printer, and demonstration of parts. [4L +10P]

4 Units of Computer CPU: memory, IO, functional diagram, concept of hardware, software and program; Operating system (OS). [6L]

5 Usage of computer: How to start and stop(under one OS), handling the mouse, keyboard, and monitor , starting and quitting a program. [3L+15P]

6 Fun with computer: typing, painting use of calculator, games. [2L+30P]

Class VII

1 Introduction (abridged version of previous syllabus): different generations of computer, components of personal computer, functional diagram, operating system, typing. [5L+9P]

2 Storage and address: concept of memory - -primary and secondary, bit, byte, capacity, address, content (numeric, ASCII characters); RAM, ROM; secondary memory – magnetic (hard disk, floppy), optional (CD, DVD): Structure and capacity: bus. [6L]

3 Classification of computer: physical size. Memory size, speed; CPU type, handling types of job; hardware issue, software issue, price; sketch of different types of computer. [6L]

4 Printer: dot-matrix printer, laser printer, inkjet printer, line printer. [2L]

5 Monitor: resolution color, size, shape. [1L]

6 OS environment (GUT based): introduction; use of mouse, icons, desktop, status, bar, start menu, selecting applications: maximizing, minimizing and restoring windows; components of a window- frame, title, menu, status, and scroll bar, using right button of mouse: changing the look and feel of desktop. [6L+9P]

7 Basic file handling: concept of directory, file and folder, viewing file and folder, operations of file and folder-creating, renaming. Deleting, copying, moving, and searching. [4L+9P]

8 Word processing: introduction, creating, editing, formatting, spell checking, And use of thesaurus, cut, copy and paste: changing color, size, font, type and alignment; saving documents, printing. [4L+9P]

9 Network and Internet: Introduction; connection through modem, WWW; browser; web page, visiting sites; e-mail [2L+2D]

10 Destructive software: virus, worm [1L]

11 Introduction to multimedia: text, audio, video. [1L+2D]

12 Linux and Windows OS: Overview, LILO boot. [4L+2D]

Class VIII

1 Introduction (abridged version of previous syllabus): different generations of computer, components of personal computer, functional diagram, operating system, Storage and address, different types of computers, GUT based OS environment, Word processing, file handling.

2 Introduction to software system classification - system(OS, compiler, interpreter, utility); programming language- procedural, non procedural, assembly and machine code; concept of flow chart, programming and problem solving; program in execution- documentation and debugging

3 Number system and computer arithmetic: Non-positional, positional (binary signed and unsigned, octal, hexadecimal, decimal); BCD; binary arithmetic- addition, subtraction, representation of negative numbers.

4 OS utilities: Control Panel- add new hardware, add/ remove program, setting date and time, desktop theme, display setting, fonts password printer setup.

5 Word processing : creation of table, addition and deletion of rows and columns, changing column width, page layout setting, margin setup, printing, mail merge, inserting header and footer, importing text and image from other software, saving document as web page.

6 Spread sheet: introduction, creating. Saving and opening a spreadsheet, entering a data, working with columns and rows, operations on cells, formatting cells, spread sheet calculations, formula basics, entering formula, creating simple charts and graphs, formatting and printing.

7 Internet and browsing: Overview, connecting through modem, dial-up and leased lines, visiting sites and pages using hyperlink, downloading and uploading.

8 E- mail: Basic concept, creating an e-mail account, sending and receiving mails.

9 Multimedia: Hardware and software requirement, playing with audio and video files using standard players.

Class IX

1 Introduction (abridged version of previous syllabus): different generations of computer, components of personal computer, functional diagram, operating system, Storage and address, different types of computers, GUT based OS environment, Word processing, spread sheet and file handling. [4L+10P]

2 Boolean algebra and logic: simple concept- truth table, AND, OR, NOT gates; simple problem solving. [4L]

3 Linux preliminaries: Introduction, GNOME desktop; file and directory- viewing, searching, changing, program execution; some simple commands – cat, cal, date, who, passwd, mkdir, rmdir, cp, mv, more, Is, Ip, tty, redirection, pipe, tree, editor. [4L+8P]

4 Presentation software: Creating new presentation, working with fonts, styles; setting colors for background and foreground, inserting sounds, pictures drawing animations, 3D objects; modifying slide layout; presentation of slide show. [6L+8P]

5 Use of data base package: Concept , feature and use of database; table ; creation of table –field, attribute of field and record; entering, viewing and modifying data; modification of table, saving and opening database files, handling multiple tables and simple queries. [15L+15P]

6 Internet and services: Overview, URL, domain name, web page and HTML, search engine, bulletin board, chat, downloading and uploading files, telnet, ftp, e-mail, operation and principle. [4L+6P]

Class X

1 Introduction (abridged version of previous syllabus): components of personal computer, functional diagram, operating system, Storage and address, different types of computers, GUT based OS environment, Word processing, spread sheet , Internet, e-mail, multimedia, presentation software database package. [12L+15P]

2 Programming: Concept and language; C preliminaries –Structure of a C program, headers, character set, identifiers, and keywords, data type, constant, declaration, arithmetic operators, unary operators, expressions, assignment statement, relational and logical operators, hierarchy of operators, library functions, input and out put functions- scanf, printf, control statement-, statemen, nested if concept of a loop- for statement, while statement, do- while statement, nested loops, switch break, continue, arrays- defining an array, multi dimensional arrays, arrays and string. [10L+15P]

3 Projects: Three Projects- One each on a)data base b)spread sheet c)C [3L+15P]

Class XI

1 Introduction to information technology and use of internet: Date and information, data processing; computer- characteristics and uses, components, I/O devices, hardware and software, system software – OS, translators(compilers, interpreters and preprocessors), linkers and loaders, application software and package- commercial, shareware, freeware. Public domain, utilizes, language- machine assembly, procedural and object oriented languages (with advantages and disadvantages), multi media- integration of text, images, audio and video files, networks- telephone, mobile, LAN,MAN, WAN, advantages and problems, TCP/IP, Internet and WWW-IP, ISP, DNS, browsers, search engines, mail, use net, chat, cookies; virus and worms. [6L+2P]

2 Web design: HTML-characteristics and uses, structure of a HTML page, texts and tags- paired and standalone, comment, structure tags- HTML , HEAD, BODY, BASE, LINK, META, heading and title- H1………H6,TITLE, paragraph-P links-A, List- OL, UL,MENU, DIR,LI,DL, DT, DD, ADDRESS, BLINK, FONT, image- IMG, HR, MRQUEE, inserting, aligning and resizing, forms- FORM, INPUT, TEXT AREA, SELECT, OPTION, ISINDEX, list box, check box and radio buttons, tables- TABLE, CVAPTION, audio and video, design of a web page, dynamic web page. [4L+8P]

3 Scripting language: purpose, writing simple scripts using a scripting language. [4L+6P]

4 Linux: History, GNU General Public License, different types of distributions, advantages and disadvantages; installation- boot disk, dual booting and LILO, daemons and services, user creation, logging in and shutting down; file and directory

Class XII

1 Computer installation: Hardware- connection of keyboard, monitor, mouse, printer, FDD, HDD, CD/DVD, network card, modem; Software- OS, browser, public domain software like Acrobat reader, other utilities.
[5L+10P]

2 Computer ethics: Principle and values; intellectual property, copyright, site license; software piracy, copy protection, hacking.
[5L]

3 Destructive codes: virus, worm, bomb, Trojan horse; vaccine, virus scan.
[3L]

4 SQL: Table creation, inserting data in tables, viewing data in tables, filtering table data, modifying structure of a table, renaming table, deleting table; computations on table set- arithmetic operation, logical operation, range searching pattern matching; SQL function numeric, string, aggregate function; joining multiple tables.
[10L+10P]

5 Projects: Two projects on any two of the following areas: a) simple database application using SQL b) web design and/or multimedia c) E- commerce d) any other relevant topic based on the curriculum.
[30P]

Note:
(i) L: Lecture Period D: Demonstration Period P: Practical Period
(ii) Some of the practical classes should be allotted for quiz, examination and practice.

viewing, searching, changing- cat, Is, grep, cd, redirection, pipe, tee, wild card, user permission; kernel- process and status; GNOME desktop- main menu, pager, task list, applets, desktop links and launchers; execution variables, positional parameter, shell operators && and II, simple shell scripts like ( ) profile, welcome message, search for multi word pattern; some simple commands (alphabetical listing) to be used with appropriate sections- alias, arch, at, cal, cat, chgrp, chmod,clear, cp, date, dir, dump, echo, emacs, find, finger, ftp, grep, gzip, kill, locate, login, logname, Is, Ip, make, man, mkdir, more, mv, newgrp, passwd, ping, ps, pwd, read, nmdir, route, set, shutdown, sleep, sty, su, touch, tr, tree, tty, user, vmstat, wall, who, whoami, xmessage, xmodmap, xset, xwininfo, zip, comparison between DOS and Linux commands commands; text editor- vim. [8L+11]

5 Programming: Concept and Language; C preliminaries- structure of a C program, headers, character set, identifiers and keywords, data type, constant, declaration, arithmetic operators, unary operators, expression, assignment statement, relational and logical operators, hierarchy of operators, library functions, input and output function – scanf, printf, control statement- if-else statement, nested if, concept of a loop, for statement, while statement, do- while statement, nested loops, switch break, continue; arrays- defining an array, multidimensional arrays, arrays and string; functions- basics, declaring a function, external variables, scope, rules, recursion; string- basic string operation; pointers- pointers and address, pointers and arrays, address arithmetic, character pointers; structures- defining a structure, arrays of structure. [12L+15P]

CBSE SYLLABUS

CLASS IX

Unit 1: IT Basics

1. Introduction
2. Generation of a computer.
3. Shapes and sizes of computer.
4. Hardware and software.
5. Anatomy of computers.
6. Storage devices.
7. Computer Languages.
8. Operating systems
9. Communication technology.
10. Information superhighway-Internet.
11. Requirement for getting Online.
12. How the Internet works.

Unit 2: IT Tools

Chapter 1 : MS- Windows.

1. Introduction
2. Getting familiar with windows environment.
    a. Features of Windows
    b. Starting and logging on to Windows.
    c. Exiting Windows and turning off your computer.
    d. Active desktop.
3. Start Menu.
4. Working with files and folders.
    a. Understanding storage devices, folders and files.
    b. Using my computer to see what is in your computer.
    c. Opening a Folder.
    d. Creating and Renaming a Folder.
    e. Copying , Moving and Deleting a Folder.
    f. Opening , Renaming and Deleting a File.
    g. Copying a File.
    h. Restoring a Deleted File and Emptying the Recycle Bin.
    i. The Find Feature.
    j. Using Windows Explorer .
    k. Working with Control Panel.
    l. Minimizing , Maximizing and Restoring a Window.
    m. Closing a Window.
5. Adding Windows Accessories
    a. Calculating with calculator.
    b. Using Notepad.
    c. Using Wordpad.
    d. Using Windows Paint.

Chapter 2 : MS-Word

1. Introduction
2. Introduction to Word Window.
    a. The Title Bar
    b. The Menu Bar
    c. The Status Bar
    d. The Ruler Bar
    e. Toolbars
    f. Scroll Bar
    g. Workspace

h. Document Text

3. Highlighting Text
4. Highlighting Menu Items.
5. Opening , Creating and Saving Documents.
6. Printing.
7. Spelling and Grammar.
8. Formatting Text and Paragraphs.
9. Paragraph Formatting.
10. Indent Paragraphs.
11. Borders and Shading.
12. Bulleted/ Numbered Lists.
13. Working with tables
14. Inserting WordArt, ClipArt and Pictures.
15. Mail Merge.

Chapter 3 : MS- PowerPoint

1. Introduction
2. Some commonly used terms in Power Point.
3. Getting Started.
4. Identifying Toolbars.
5. The different views of Power Point.
6. Creating of a Presentation.
7. Adding Graphics to your Presentation.
8. Slide Master.
9. Template.
10. Animation Effects
11. Action buttons.
12. Adding Videos and Animated GIF pictures.
13. Transitions and Build Slides.
14. Automating Presentation.
15. Running a Slide Show.

Chapter 4 : MS- Excel

1. Introduction
2. Getting started with Excel.
3. Toolbars.
4. Starting and Exiting Excel.
5. Saving Workbook.
6. Printing.
7. Formatting and Customizing data.
8. Formatting Numbers.
9. Functions and Formulas.
10. Other commonly used functions
11. Array Formulas.
12. Totaling Rows and Columns automatically.
13. Naming Cells and Renges.
14. Tables.
15. Using Tables.
16. Creating and Editing Charts.

Unit 3 : IT Applications

1. Microsoft Word Project.
2. Microsoft PowerPoint Project.
3. Microsoft Excel Project.

CLASS X

Unit 1 : Internet

1. Concept of Internet
2. Modes of connecting to Internet.
3. WEB browsers.
4. WEB Servers, WEB Sites and WEB Pages.
5. WEB Address.
6. Electronic Mail
7. Search Engines.
8. Visiting WEB Sites.
9. Internet Relay Chat.
10. Video conferencing.
11. File transfer protocol.
12. Downloading.
13. Usenet and Newsgroups
14. Telnet.

Unit 2 : IT Tools

Chapter 1 : Microsoft Access

1. Concepts of Database
2. Data Base Management Systems.
3. Major Advantages of DBMS.
4. Uses of Data Base Management Systems.
5. Introduction to MS-Access.
6. Components of MS-Access.
7. Starting MS-Access.
8. Opening an existing database.
9. Creating a table using Table Wizard.
10. Design view of a table.
11. Data types.
12. Input Mask
13. Field properties.
14. Adding a Field in the Table.
15. Inserting a New Field in Database View.
16. Deleting a wrongly inserted field.
17. Inserting and deleting a field in design view.
18. Renaming a field in design view.
19. Renaming a field in datasheet view.
20. Components of design view.
21. Entering data.
22. Deleting a field from a table.
23. Deleting field in data sheet view.
24. Deleting data in design view.
25. Hiding columns.
26. Creating a primary key.
27. How to remove Primary Key.
28. Manipulation of Data.
29. Replacing Data.
30. Shortcut buttons for manipulating data.
31. Deleting Records.
32. Toolbars and views of table.
33. Column width and Row height.
34. Increasing / Decreasing the size of fields.
35. Crating relationship between tables.

36. Adding OLE object(s) in table.
37. Creating and Editing reports.

Chapter 2 :Hypertext Markup Language.

1. Concept of a WEB browser.
2. Internet Explorer.
3. Netscape Navigator.
4. Introduction to HTML.
5. Characteristics of HTML and Importance of HTML.
6. WEB page Designing using HTML.
7. Elements of HTML.
    a. Head Element.
    b. Title Element
    c. Body Element.
    d. P Element.
    e. Font Element.
    f. BR and HR Element.
    g. Heading Element.
    h. Basefont.
    i. Center.
    j. Align.
8. Character Entities.
9. Overview of WEB Page Designing.
10. Creating lists.
11. Inserting images.
12. Tables.
13. Linking WEB Pages.
14. Designing Forms.
15. Frames.

Unit 3 : IT Applications

1. Projects on Database.
2. Projects on WEB designing.

CLASS XI (COMPUTER SCIENCE)

Unit 1 : Computer Fundamental

1. Evolution of computers
2. Basics of computer and its operation
3. Functional Components and their inter connections
4. Concept of Booting
5. Use of Operating System for directory listing
6. Hierarchical directory structure.
7. Renaming and deleting files /folders
8. Formatting floppy
9. Copying files
10. Concept of Path and Path name.
11. Switching between tasks
12. Installation / removal of applications

Unit 2 : Software Concepts

1. Types of software-System Software, Utility Software and Application Software
2. System software- Operating System , Compilers , Interpreters and Assemblers
3. Operating System – Need for operating system, Functions of OS
4. Types of OS
5. Commonly used OS

Unit 3 :Programming Methodology

1. General Concepts
2. Modular approach
3. Clarity and Simplicity of Expressions
4. Use of proper names for identifiers.
5. Comments and Indentation.
6. Documentation and Program Maintenance.
7. Running and Debugging programs.
8. Syntax ,Run time and Logical Errors.
9. Problem solving Methodology and Techniques-
   a. Understanding of the problem
   b. Identifying minimum number of inputs required for output.
   c. Step by Step solution for the problem
   d. Breaking down solution into simple steps
   e. Identification of arithmetic and logical operations required for solution.
10. Using control structure- Conditional control and looping.

Unit 4 : Introduction to Programming in C++

Programming by Example in C++ Language :

1. C++ character set
2. C++ Tokens (Identifiers, Keywords, Constants, Operators)
3. Structure of a C++ Program (include files, main function); Header files – iostream.h, iomanip.h, cout, cin;
4. Use of I/O operators (<< and >>)
5. Use of endl and setw()
6. Cascading of I/O operators
7. Error Messages
8. Use of editor, basic commands of editor
9. Compilation, linking and execution;
10. Standard input/output operations from C language: gets(), puts() of stdio.h header file

Data Types, Variables and Constants:

1. Concept of Data types
2. Built-in Data types: **char**, **int**, **float** and **double**
3. Constants: Integer Constants, Character Constants (Backslash character constants - \n, \t), Floating Point Constants,String Constants; Access modifier: **const**; Variables of built-in data types,
4. Declaration/Initialisation of variables, Assignment statement; Type modifier: signed, unsigned, long;

Operators and Expressions:

1. Operators: Arithmetic operators (-,+,*,/,%), Unary operator (-), Increment and Decrement Operators (—,++), Relational operators (>,>=,<,<=,==,!=), Logical operators (!, &&, ||), Conditional operator: <condition>?<if true>:<else>; Precedence of Operators;
2. Expressions; Automatic type conversion in expressions, Type casting; C++ shorthand's (+=, -=, *=, /=, %=);

Flow of control:

1. Conditional statements: **if-else**, Nested **if**, **switch..case..default**, Nested **switch..case**, break statement (to be used in switch..case only);
2. Loops: **while**, **do - while** , **for** and Nested loops;

Structured Data Type: Array

1. Declaratrion/initialisation of One-dimensional array,
2. Inputting array elements,
3. Accessing array elements
4, Manipulation of Array elements (sum of elements, product of elements, average of elements, linear search, finding maximum/minimum value);
5.Declaration/Initialization of a String
6.String manipulations (counting vowels/consonants/digits/special characters, case conversion, reversing a string, reversing each word of a string);

String Functions:

Header File: string.h
Function: **isalnum(), isalpha(), isdigit(), islower(), isupper(), tolower(), toupper();**

Character Functions:

Header File: ctype.h
Functions: **isalnum(), isalpha(), isdigit(), islower(), isupper(), tolower(), toupper(), strcpy(),strcat(), strlen(), strcmp(), strcmpi();**

Mathematical Functions:

Header File-math.h, stdlib.h;
Functions: **fabs(), log(), log10(), pow(), sqrt(), sin(), cos(), abs(),**

Other Functions:

Header File- stdlib.h;
Functions: **randomize(), random();**

Two-dimensional Array:

1. Declaration/initialisation of a two-dimensional array,
2. Inputting array elements
3. Accessing array elements,
4. Manipulation of Array elements (sum of row element, column elements, diagonal elements, finding maximum/minimum values);

User Defined Functions:

1. Defining a function;
2.Function prototype,
3 Invoking/calling a function
4. Passing arguments to function, specifying argument data types, default argument, constant argument, call by value,call by reference, returning values from a function, calling functions with arrays,
5.Scope rules of functions and variables; local and global variables;

Unit 5 :Computer System Organization

1.Number System: Binary, Octal, Decimal, Hexadecimal and conversion between two different number systems.
2. Integer
3. Floating Point,
4. 2's complement of number from base-2;
5.Internal Storage encoding of Characters: ASCII, ISCII (Indian scripts Standard Code for Information Interchange), and UNICODE;
6.Microprocessor: Basic concepts, Clock speed (MHz,GHz), 16 bit, 32 bit, 64 bit processors; Types – CISC, RISC;
7.Concept of System Buses, Address bus, Data bus,
8.Concepts of Accumulator,
9.Instruction Register, and Program Counter;
10.Commonly used CPUs and CPU related terminologies:
11.Intel Pentium Series, Intel Celeron, Cyrix, AMD Series, Xeon, Intel Mobile, Mac Series; 12.CPU Cache;
13.Concept of heat sink and CPU fan
14. Motherboard; Single, Dual and Multiple processors;
15.Types of Memory: Cache (L1,L2), Buffer, RAM (DRAM, SDRAM, RDRAM, DDRAM), ROM (PROM, EPROM)
16. Hard Disk Drive, Floppy Disk Drive, CD/DVD Drive
17. Access Time;
18.Input Output Ports/Connections: Power connector, Monitor Socket, Serial (COM) and Parallel (LPT) port, Universal Serial Bus port, PS-2 port, SCSI port, PCI/MCI socket, 19.Keyboard socket,
20.Infrared port (IR), audio/speaker socket, Mic socket; data Bus
21.External storage devices connected using I/O ports;
22.Keyboards: QWERTY, Inscript, Multilingual, Dvorak
23.Printers: Dot Matrix Printer, Line Printer, Deskjet/Inkjet/Bubblejet Printer, Laser Printer;
24.Power Supply: Switched Mode Power Supply (SMPS): Elementary Concept of Power Supply:Voltage, Current, Power (Volt, Ampere, Watt), SMPS supplies – Mother Board,
25.Power Conditioning Devices: Voltage Stabilizer, Constant Voltage Transformer (CVT),Uninterrupted Power Supply (UPS)-Online and offline.

CLASS-XI (INFORMATICS PRACTICES)

Unit 1

Chapter 1 :Computer System and Business Applications

1. Evolution of computers
2. Basics of computer and its operation
3. Functional Components and their inter-connections
4. Concept of Booting
5. Hardware concepts:

   Diagram illustrating main parts of computers;
   Central Processing Unit (CPU): Arithmetic Logic Unit (ALU), Control Unit, Memory Unit (RAM - Random Access Memory & ROM - Read Only Memory)
   Role of Input, Processing and Output Devices in a computer system
   input devices: Keyboard, Mouse, Light pen, Touch Screens, Graphics Tablets, Joystick, Mic, MICR,OCR, Scanner, Smart Card reader, Barcode reader, Biometric sensor, web camera, digital camera;Output Devices: Monitor/Visual Display Unit (VDU), Printer (Dot Matrix Printer, Desk jet/ Ink jet/ Bubble jet Printer, Laser Printer), Plotter, Speaker, Secondary Storage Devices: Floppy Disk, Hard Disk, Compact Disk, Magnetic Tape, Digital Video Disk (DVD), Zip Drive; Units of Memory: Bit (Binary Digit), Byte, Kilobyte, Megabyte, Gigabyte.
6. Software Concepts:

   Types of Software: System Software, Utility Software and Application Software.
   System Software: Operating System, Language Compilers, Interpreters and Assembler;
   Operating System: Need of operating systems, Functions of Operating System ,Types of Operating system.
7. Utility Software:

   Compression tools, Anti Virus, File Management tools and Disk Management tools;
8. Application Software as a tool:

   Word Processor, Presentation Tool, Spreadsheet Package, Database Management System; Business software (for example: Inventory Management System,Payroll System, Financial Accounting, Hotel Management, and Reservation System);
9. Development of programming languages :

   Machine Language, Assembly Language, High Level Language (BASIC, COBOL, FORTRAN, PASCAL, C++); GUI based languages - Visual Basic,Visual C++; C#, Java, vb.net.

Chapter 2 : GUI Operating System : WINDOWS

1. General features
2. Elements of Desktop – Taskbar, Icon, Start button, Shortcuts, Folder, Recycle Bin, My Computer
3. Start Menu: Program, Documents, Settings, Find/Search, Help, Run, Shut Down/Logoff
4. Customization of Taskbar, start menu, Display properties (Wallpaper, Font Settings, Color Settings, Screen Savers);
5. Program Menu: Accessories - Calculator, Notepad, Paint, Word pad, Entertainments (CD Player,Sound Recorder, Media Player, Volume Controller);
6. Internet Browsers – Mozila Firefox, Internet Explorer, Netscape Navigator.
7. Control Panel: Add new hardware; Add new Software, Printer Installation, Date/Time, Mouse,and Regional Settings;

Chapter 3 :Documentation

1. Purpose of using word processing software
2. Opening a new/existing document, closing a document, typing in a document, saving a document, print preview, printing a document, setting up of page as per the specifications, selecting a portion of document, copying selected text, cutting selected text, pasting selected text; changing font, size, style, color of text; Inserting symbol; Formatting: Alignment – Left, Right, Center; Justification;

Chapter 4 :Industries and Business Computing:

Types of Industries (Production, Shipping, Travel, Hotel, Insurance, Construction, Automobile), Applications of Business Computing in Industries.

Unit 2 : Introduction to Programming

Chapter 1 : Programming Methodology:

1. General Concepts
2. Modular approach;
3. Stylistic Guidelines: Clarity and Simplicity of Expressions,
4. Names
5. Comments
6. Indentation
7. Documentation and Program Maintenance
8. Running and Debugging programs
9. Syntax Errors, Run-Time Errors, Logical Errors;
10. Problem Solving Methodology and Techniques
11. Understanding of the problem,
12. Identifying minimum number of inputs required for output,
13. Step by step solution for the problem
14. Breaking down solution into simple steps, Identification of arithmetic and logical operations required for solution
15. Using Control Structure: Conditional control and looping (finite and infinite);

Chapter 2: Programming Tool : Visual Basic

1. Introduction to Programming – Modular Programming, Object Oriented Programming, Event Driven Programming
2. About Visual Basic (Object Based Programming Language)
3. Rapid Application Development using Visual Basic
4. Concept of Project in Visual Basic
5. VB Project Options - Standard EXE, ActiveX DLL, ActiveX EXE, ActiveX Control, ActiveX Document DLL, ActiveX Document EXE Addin, VB Application Wizard, IIS Application, DHTML Application;
3. Getting Familiar with Visual Basic User Interface - Pull-Down menus, Toolbar, Toolbox, Project Explorer, Properties Window, Form Layout window, Form, Immediate window; Opening and Closing windows, Resizing and moving windows, Docking windows; Quitting Visual Basic;
4. Visual Basic Tool Box (Standard Window Controls) - Pointer, Picture Box, Label, Text Box, Frame,Command Button, Check Box, Option Button, Combo Box, List Box, Horizontal Scrollbar, Vertical Scrollbar, Timer, Drive List box, Directory List box, File List box, Shape, Line, Image, Data, OLE
5. Object Naming Conventions
6. Event Procedures;

Chapter 7 :Programming Fundamentals

1. Data Types: Integer, Long, Single, Double, Currency, String, Byte, Boolean, Date, Object, Variant;
2. Variables: Need to use variable, Declaring Variables, Variable Naming Convention, Assigning value to Variables, Data Types of variable, Scope and lifetime of Variables (Public and Private);
3. Control Structures:
   Decision Structure – IF, IF-THEN-ELSE, Select Case;
   Looping Structure- Do While…Loop, Do…Loop While, For…Next, For Each…Next;
4. Concept of Menus, Shortcut menus and Popup menus Designing Menu System,
5. Menu Editor Dialog Box Options (Name, Index, Shortcut, Help Context ID, Negotiate Position Checked, Enabled, Visible, Window List, Right Arrow, Left Arrow, Up Arrow, Down Arrow, Menu List, Next, Insert, Delete, OK, Cancel),

6. To Create Menu Controls in the Menu Editor, Menu Naming Conventions, Setting the Name Property, Creating a Menu Control Array, Creating Sub Menus, Separating Menu Controls, Assigning Access Keys and Shortcut Keys, Controlling Menus at Runtime-Enabling and Disabling Menu Commands, Displaying a Checkmark on a Menu Control, Making a Menu Control Invisible, Adding Menu Control at Runtime, Displaying Pop-Up Menu;
7. General Controls (Advance): Image List, Common Dialog Box, ADO DC, DB Combo, Media Player Control, DB Grid
8. Adding a Toolbar: Creating an Image List, Adding Images to the Toolbar, To Add Code for the Toolbar Buttons;
9. Adding Status Bar: Adding Status Bar panels, Adding Time on the panel.
10. Dialog Boxes: Pre-defined dialog box, Custom dialog box;

UNIT 3: Relational Database Management System

Chapter 1 :Database Management System

1. Introduction to database concepts: relation/Table, attribute, Tuple / Rows, field, Data, Conceptof String, Number and Date values, Data type and Data Integrity (Domain and Referential Integrity). Candidate key, Alternate key, Primary Key, Foreign Keys; Data Normalization-first, second, third, BCNF normal form;
2. Examples of Commercially available Database Management System's (Back-End) – Oracle,MS-SQL Server, DB2, MySQL, Sybase, INGRES.
3. Examples of Front End Software's: Oracle Developer, Visual Basic, Visual C++, Power Builder,Delphi;

Chapter 2 : RDBMS Tool :

1. Interface with Oracle, Login Screen, Entering Name and Password; Classification of SQL
2. Statements: DML (SELECT, INSERT, UPDATE, DELETE), DDL (CREATE, DROP, ALTER,RENAME, TRUNCATE), DCL (GRANT, REVOKE), TCL (COMMIT, ROLLBACK);
3. SQL SELECT Statement: SQL SELECT statement, Selecting All the Columns, Selecting Specific Column, Column Heading Default, Using Arithmetic Operators, Operator Precedence,Significance of NULL value, NULL values in Arithmetic Expressions, Defining and using Column Alias, Concatenation Operator (||), Duplicate rows and their elimination (DISTINCT keyword), Role of SQL and SQL*Plus in interacting with RDBMS, Displaying Table Structure(DESC command);
4. SELECT Statement Continued: Limiting Rows during selection (using WHERE clause), Working with Character Strings and Dates,
5. Using Comparison operators, BETWEEN Operator, IN Operator, LIKE Operator, IS NULL Comparison, Logical Operators, Use of Logical Operators (AND/OR/NOT Operators), Logical Operator Precedence,
6. ORDER BY Clause, Sorting in Ascending/Descending Order, Sorting By Column Alias Name, Sorting On Multiple Columns;
7. Functions: SQL Functions, Types of SQL Function (Single Row/Multiple Row),
8. Single Row SQL Functions, Character Functions (Case Conversion/Character Manipulation), Case Conversion Functions (lower (), InitCap (), UPPER ()) Character Manipulation Function (CONCAT(), INSTR(), LENGTH(), TRIM(), SUBSTR(), LPAD()), Number Functions (ROUND(), TRUNC(), MOD()), Working with Dates (LAST_DAY(), MONTHS_BETWEEN(),NEXT_DAY(), ADD_MONTHS(), ROUND(), TRUNC()) Arithmetic Operation on Dates, Date Functions and their Usage, Data type Conversion Functions, Implicit and Explicit Conversion,TO_CHAR Function with Dates, TO_CHAR Function For Numbers, TO_NUMBER and TO_DATE Functions, NVL Function and its Usage, DECODE Function and its Usage;

9. Grouping Records: Concept of Grouping Records and Nested Grouping, Nested Grouping of records, Group Functions, Types of group functions (MAX(), MIN(), AVG(), SUM(), COUNT()),

10. Using AVG and SUM Functions, Using MIN and MAX Functions, Using the COUNT Function,using COUNT(*), DISTINCT clause with COUNT, Group Functions and Null

Values, Using NVL Function with Group Functions,

11. Grouping Records: Group By Clause, Grouping By More than One Column, Illegal Queries with Group By Clause, Excluding Group Results: Having
12. Clause, Nesting Group Functions,
13. Sub Queries: Concept of Sub-Query, Sub Query to Solve a Problem, Guidelines for Using Sub Queries, Types of Sub-Queries (Single Row and Multiple Row) and (Single Column and Multiple Column); Single Row Sub-Query and its Execution;
14. Displaying Data From Multiple Tables: Concept of Join, Result of Join, Cartesian Product and Generating Cartesian Product example using Mathematical Set), Types Of Joins (EQUI, SELF,NON-EQUI, OUTER (LEFT and RIGHT)), Equi-join: RetrievingRecords with Equi-join,Additional Search Conditions using AND operator, Short Naming Convention for Tables (Table Aliases), Non-Equi join and its Implementation, Outer-Join and Its Usage, Self-Join (Joining a table to Itself);
15. Manipulating Data of A Table/Relation: Concept of DML (Data Manipulation Language),INSERT Statement, Inserting New Rows, Inserting New Rows with Null Values, Inserting Date Values, Use of Substitution Variable to Insert Values, Copying Rows From Another Table, Update Statement to Change Existing Data of a Table, Updating Rows In A Table, Updating Rows Based on Another Table, Delete statement/ Removing Row/Rows from a Table, Deleting Rows Based on condition from another Table; Making Data Manipulation Permanent (COMMIT). Undo Data Manipulation Changes (ROLLBACK)
16. Database Objects: View, Table, Sequence, index, and Synonyms, DDL (Data Definition Language), Naming Convention, Creating Views, Creating Synonyms, Simple Views and Complex Views, Retrieving Data From a View, Querying a View, Modifying a View.
17. Including Constraints: Constraints, Concept of using Constraints, Constraint Guidelines, Defining Constraints, NOT NULL, UNIQUE KEY, PRIMARY KEY, FOREIGN KEY, FOREIGN KEY Constraint Keywords, CHECK, Adding a Constraint, Dropping a Constraint, Disabling Constraints, Enabling Constraints, Viewing Constraints, Viewing The Columns, Associated with Constraints;
18. Creation of a Table/Relation: CREATE TABLE Statement, Data types, The DEFAULT option, Creating Tables, Referencing Another User's Tables, Querying the Database Dictionary to view all tables in the Oracle Database, Creating a Table by Using a Sub-Query;
19. Managing Existing Tables and other Database Objects: The ALTER TABLE Statement, Adding a New Column in a Table, Modifying Existing Column, Dropping a Column, Renaming an Object, Truncating a Table, Adding Comments to a Table, Dropping Views, Dropping Synonyms, Dropping Tables; giving permission to other users to work on Created Tables and Revoking it (GRANT and REVOKE statement).

CLASS XI (MULTIMEDIA AND WEB TECHNOLOGY)

Unit 1: Computer System

Chapter 1 :Introduction to Computer

1. Input Devices – Keyboards, Mouse, Joy stick, Mic, Camera;
2. Output Devices – Monitor, Printer, Speaker, Plotter;
3. Memory Units – Byte, Kilobyte, Megabyte, Giga byte,tera byte; Primary Memory – RAM and ROM; Secondary Storage devices – Floppy Disk, Hard disk, CD ROM, DVD, Zip Drive, DAT Drive; Power devices – UPS; Software – System Software,
4. Application Software, Utility Software;
5. Working on computers – switching on computer, booting computer; icons, shortcuts, taskbar, mouse pointer; typing, saving and printing a simple text file,drawing simple picture using MSPaint, using calculator option, customizing desktop, windows explorer, managing folders (creating, moving, deleting, renaming); using floppy disk drive, using CD/DVD drives; managing files (copying, moving, deleting, renaming); playing audio and video;

Chapter 2 : GUI Operating System (Windows)

1. General features, Elements of Desktop - Taskbar, Icon, Start button, Shortcuts, Folder, Recycle Bin, My Computer;
2. Start Menu: Program, Documents, Settings, Find/Search, Help, Run, ShutDown/Logoff;
3. Customization of Taskbar, Start menu, Display properties (Wallpaper, Font Settings, Color Settings, Screen Savers);
4. Program Menu: Accessories - Calculator, Notepad, Paint, Word pad, Entertainments (CD Player,Sound Recorder, Media Player, Volume Controller);
5. Browsers: Mozila Firefox, Internet Explorer, Netscape Navigator;
6. Control Panel: Add new hardware; Add new Software, Printer Installation, Date/Time, Mouse, and Regional Settings;

Chapter 3 : Documentation

1. Purpose of using word processing software,
2. Opening a new/existing document,
3. Closing a document,
4. Typing in a document
5. Saving a document
6. Print preview, printing a document
7. Setting up of page as per the specifications
8. Selecting a portion of document
9. Copying selected text,
10. Cutting selected text, pasting selected text
11. Changing font, size, style, color of text;
12. Inserting symbol
13. Formatting: Alignment – Left, Right, Center; Justification;

Unit 2 : Web Development

Chapter 1

1. Web Pages
2. Hyper Text Transfer Protocol (HTTP)
3. File Transfer Protocol (FTP) Domain Names
4. URL
5. Protocol Address
6. Website
7. Web browser
8. Web Servers
9. Web Hosting.

Chapter 2 :HTML/DHTML

1. Introduction,
2. Objectives
3. Introduction to Universal Resource Identifier (URI) - Fragment Identifiers and Relative URI'
4. History of HTML, SGML
5. Structure of HTML/DHTML Document
6. Switching between opened Windows and browser (Container tag, Empty tag, Attribute);
7. Basic Tags of HTML: HTML, HEAD, TITLE, BODY (Setting the Fore color and Background color, Background Image, Background Sound), Heading tag (H1 to H6) and attributes (ALIGN),FONT tag and Attributes (Size: 1 to 7 Levels, BASEFONT, SMALL, BIG, COLOR), P, BR,
8. Comment in HTML (<! >), Formatting Text (B, I, U, EM, BLOCKQUOTE, PREFORMATTED,SUB, SUP, STRIKE), Ordered List- OL (LI, Type- 1, I, A, a; START, VALUE), Unordered List- UL (Bullet Type- Disc, Circle, Square, DL, DT, DD), ADDRESS Tag;
9. Creating Links: Link to other HTML documents or data objects, Links to other places in the same HTML documents, Links to places in other HTML documents;

   Anchor Tag <A HREF> and <A NAME>, Inserting Inline Images <IMG ALIGN, SRC, WIDTH,HEIGHT, ALT, Image Link, Horizontal Rules <HR ALIGN, WIDTH, SIZE, NOSHADE>;

Chapter 3 : Web Page Authoring Using HTML

1. Tables: Creating Tables, Border, TH, TR, TD, CELLSPACING, CELLPADDING, WIDTH,COLSPAN, CAPTION, ALIGN, CENTER;
2. Frames: Percentage dimensions, Relative dimensions, Frame – Src, Frameborder, height and width, Creating two or more rows Frames <FRAMESET ROWS >, Creating two or more Columns Frames <FRAMESET COLS >, <FRAME NAME SRC MARGINHEIGHT MARGINWIDTH SCROLLING AUTO NORESIZE>, <NOFRAMES>, </NOFRAMES>;
3. Forms: Definition, Use – Written to a file, Submitted to a database such as MSAccess or Oracle,E-mailed to someone in particular, Forms involve two-way communication;
4. Form Tags: FORM, <SELECT NAME, SIZE, MULTIPLE / SINGLE> <OPTION> .. </SELECT>, <TEXTAREA NAME ROWS COLS > , , </TEXTAREA>, METHOD,CHECKBOX, HIDDEN, IMAGE, RADIO, RESET, SUBMIT, INPUT <VALUE, SRC,CHECKED, SIZE, MAXLENGTH, ALIGN>;

Chapter 4 : Document Object Model

1. Concept and Importance of Document Object Model,
2. Dynamic HTML documents and DocumentObject Model.
3. Cascading Style Sheets
4. Introduction to Cascading Style Sheet (CSS), three ways of introducing the style sheets to your document. Basic Syntax; Creating and saving cascading style sheets. <STYLE> tag.Examples showing the linking of external style sheet files to a document; Inline and Embed, <DIV>tag; COLOR, BACKGROUND-COLOR, FONT-FAMILY, FONT-STYLE, FONT-SIZE and FONTVARIANT; FONTWEIGHT, WORD-SPACING, LETTER-SPACING, TEXTDECORATION, VERTICAL-ALIGN, TEXT-TRANSFORM; TEXT-ALIGN, TEXT-INDENT, LINEHEIGHT,
5. Introduction to Margin, Padding and Border;MARGINS (all values), MARGIN-PROPERTY, PADDIND (all values), PADDINGPROPERTY;BORDER (all values), BORDER-PROPERTY, BACKGROUNDIMAGE,BACKGROUNDREPEAT; Additional Features, Grouping Style Sheets, Assigning Classes;Introduction to Layers, <LAYER>, <ILAYER> tag;

Chapter 5 :eXtensible Markup Language (XML)

1. XML: Introduction;
2. Features of XML: XML can be used with existing protocols, Supports a wide variety of applications, Compatible with SGML, XML documents are reasonably clear to the layperson;
3. Structure of XML: Logical Structure, Physical Structure;
4. XML Markup: Element Markup i.e (<foo>Hello</foo>), Attribute Markup i.e.

(<!element.name property="value">) ;

5. Naming Rules: used for elements and attributes, and for all the descriptors, Comments
6. Entity Declarations :<! ENTITY name "replacement text">;
7. Element Declarations: <!ELEMENT name content>;
8. Empty Elements: <!ELEMENT empty.element EMPTY>;
9. Unrestricted Elements: <!ELEMENT any.element ANY>;
10.Element Content Models : Element Sequences i.e. <!ELEMENT counting(first, second, third, fourth)>,Element Choices <!ELEMENT choose(this.one | that.one)>, Combined Sequences and Choices;
11.Element Occurrence Indicators :-Discussion of Three Occurrence Indicators
   ? (Question Mark)
   * (Asterisk Sign)
   + (Plus Sign)
12.Character Content: PCDATA (Parseable Character data) <!ELEMENT text(#PCDATA), Document Type Declaration (DTD) and Validation;
13. Developing a DTD: Modify an existing SGML DTD, Developing a DTD from XML Code, either automatically or manually;
14. Viewing XML in Internet Explorer, Viewing XML using the XML Data Source Object.
15. XSL (Extensible Style Sheet Language) or CSS (Cascading Style Sheet);

Unit 3: Web Scripting

Chapter 1: VBScript

1. Introduction
2. Adding VBScript code to HTML page
3. VBScript Data type-Variant subtypes,
4.VBScript Variables: (Declaring variable, Naming restrictions, Assigning value to variables, Scalar variables and 1-D Array),
5.VBScript Constants,
6.VBScript Operators, and Operator precedence;
7.MsgBox: functions of message box (Prompt, Buttons, Title, Helpline, Context), Return values of MsgBox function, button argument setting.
8.Conditional statements: If..Then.. Else, Select case;
9.Loops: Do loops, While.. Wénd, For.. Next, For..Each..Next;
10.VBScript variables: Sub procedures, Function procedures;
11.Using VBScript with HTML form controls, Data handling functions, String functions, Dateand Times functions;

Unit 4:

Chapter 1 :Multimedia And Authoring Tools

1. Graphics Devices: Monitor display configuration, Basics of Graphics Accelerator Card and its importance;
2. Basic concepts of Images: Digital Images and Digital Image Representation
3. Image Formats :TIFF, BMP, JPG/JPEG, GIF, PIC. PDF, PSD;
4. Theory of design, form, line, space, texture, color, typography, layout, color harmony, unity, balance, proportion, rhythm, repetition, variety, economy, still life, light and shade, Poster Design;Still life, colored layout, Poster Design, Designing of Books, magazines brochures, children's literature, narrative text handling, scripts in Indian Languages, picture books, comics, illustrations with photographs, scientific illustrations, conceptual illustrations, handling of assignment for the market;

5. Image Scanning with the help of scanner: Setting up Resolution, Size, File formats of images;image preview, Bitonal, Grey Scale and color options; Significance of PDF-creation, modification; Animation, Morphing and Applications
6. Graphic Tools: Image Editing Software (Photoshop / Coreldraw)
7. Basic Concepts: An Introduction, creating, Opening and saving files, Menus, Toolbox, Color control icons, Mode control icons, Window controls icons; creating new images,

Image capture (TWAIN) from scanner other files;

8. Image Handling: Cropping an image, adjusting image size, increasing the size of the work canvas, saving an image;
9. Layers: Adding layers, dragging and pasting selections on to layers, dragging layers between files, viewing and hiding layers, Editing layers, rotating selections, scaling an object, preserving layers transparency, moving and copying layers, duplicating layers, deleting layers, merging layers, using adjustment layers;
10. Channels and Masks: Channel palette, showing and hiding channels, splitting channels in to separate image, merging channels, creating a quick mask, editing masks using quick mask mode;
11. Painting and Editing: Brushes palette, brush shape, creating and deleting brushes, creating custom brushes, setting brush options, saving, loading and appending brushes, Options palette; Opacity, pressure, or exposure, paint fade-out rate, making selections, using selection tools,adjusting selections, softening the edges of a selection, hiding a selection border, moving andcopying selections, extending and reducing selections, pasting and deleting selections, Image tracing (CorelDraw).
12. Concept of Multimedia: Picture/Graphics, Audio, Video;
13. Sound: Recording Sound using Sound Recorder (Capture), Sound capture through sound editing software (ex: Sound forge), Sound editing, Noise correction, Effect enhancement ; Voice Recognition Software Philips/Dragon, MIDI Player, Sound Recorder, MONO & Stereo. Sound File Format: AIFF (Audio Input File Format from Apple Mac) , MIDI, WAV, MP3, ASF(Streaming format from Microsoft).
    Importing audio and saving audio from Audio CD.
14. Sound Quality: CD Quality, Radio Quality, Telephone Quality;

CLASS XII (COMPUTER SCIENCE)

UNIT 1: PROGRAMMING IN C++

REVIEW: C++ covered In Class -XI,
Defining a symbol name using typedef keyword and defining a macro using #define directive;Need for User defined data type;

Structures:

Defining a Structure, Declaring structure variables, Accessing structure elements, Passing structure to Functions as value and reference argument/parameter, Function returning structure, Array of structures, passing an array of structure as an argument/ a parameter to a function;

Object Oriented Programming:

Concept of Object Oriented Programming – Data hiding, Data encapsulation, Class and Object,Abstract class and Concrete class, Polymorphism (Implementation of polymorphism usingFunction overloading as an example in C++); Inheritance, Advantages of Object Oriented Programming over earlier programming methodologies,

Implementation of Object Oriented Programming concepts in C++:

Definition of a class, Members of a class - Data Members and Member Functions (methods),
Using Private and Public visibility modes, default visibility mode (private); Member function
definition: inside class definition and outside class definition using scope resolution operator
(::); Declaration of objects as instances of a class; accessing members from object(s), Array of type class,
Objects as function arguments - pass by value and pass by reference;

Constructor and Destructor:

Constructor: Special Characteristics, Declaration and Definition of a constructor, Default
Constructor, Overloaded Constructors, Copy Constructor, Constructor with default arguments;
Destructor: Special Characteristics, Declaration and definition of destructor;
Inheritance (Extending Classes): Concept of Inheritance, Base Class, Derived Class, Defining derived classes,
protected visibility mode; Single level inheritance, Multilevel inheritance and Multiple inheritance, Privately
derived, Publicly derived and Protectedly derived class,accessibility of members from objects and within
derived class(es);

Data File Handling:

Need for a data file, Types of data files – Text file and Binary file;
Basic file operations on text file: Creating/Writing text into file, Reading and Manipulation of text from an
already existing text File (accessing sequentially);
Binary File: Creation of file, Writing data into file, Searching for required data from file,
Appending data to a file, Insertion of data in sorted file, Deletion of data from file,Modification of data in a file;
Implementation of above mentioned data file handling in C++;
Components of C++ to be used with file handling:
Header file: fstream.h; ifstream, ofstream, fstream classes;
Opening a text file in **in**, **out**, and **app** modes;
Using cascading operators for writing text to the file and reading text from the file; **open()**, **get()**,**put()**, **getline()**
and **close()** functions; Detecting end-of-file (with or without using **eof()** function);Opening a binary file using
in, **out**, and **app** modes;
open(), **read()**, **write()** and **close()** functions; Detecting end-of-file (with or without using **eof()**function);
tellg(), **tellp()**, **seekg()**, **seekp()** functions

Pointers:

Declaration and Initialization of Pointers; Dynamic memory allocation/deallocation operators:**new, delete;**
Pointers and Arrays: Array of Pointers, Pointer to an array (1 dimensional array),Function returning a pointer,
Reference variables and use of alias; Function call by reference.Pointer to structures: Deference operator: *, ->;
self referencial structures;

UNIT 2: DATA STRUCTURES

Arrays:

One and two Dimensional arrays: Sequential allocation and address calculation;
One dimensional array: Traversal, Searching (Linear, Binary Search), Insertion of an element in an array, deletion of an element from an array, Sorting (Insertion, Selection, Bubble sort), concatenation of two linear arrays, merging of two sorted arrays;
Two-dimensional arrays: Traversal, Finding sum/difference of two NxM arrays containing numeric values, Interchanging Row and Column elements in a two dimensional array;

Stack (Array and Linked implementation of Stack):

Operations on Stack (PUSH and POP) and its Implementation in C++, Converting expressions from INFIX to POSTFIX notation and evaluation of Postfix expression;

Queue: (Circular Array and Linked Implementation):

Operations on Queue (Insert and Delete) and its Implementation in C++.

UNIT 3: DATABASES AND SQL

Database Concepts:

Relational data model: Concept of domain, tuple, relation, key, primary key, alternate key, candidate key;
Relational algebra: Selection, Projection, Union and Cartesian product;

Structured Query Language:

General Concepts: Advantages of using SQL, Data Definition Language and Data Manipulation Language;
Data types: NUMBER, CHARACTER, DATE;
SQL commands:
CREATE TABLE, DROP TABLE, ALTER TABLE, UPDATE...SET..., INSERT, DELETE;
SELECT, DISTINCT, FROM, WHERE, IN, BETWEEN, GROUP BY, HAVING, ORDER BY;
SQL functions: SUM, AVG, COUNT, MAX and MIN;

UNIT 4: BOOLEAN ALGEBRA

Binary-valued Quantities, Boolean Variable, Boolean Constant and Boolean Operators: AND,OR, NOT; Truth Tables; Closure Property, Commutative Law, Associative Law, Identity law,Inverse law, Principle of Duality, Idem potent Law, Distributive Law, Absorption Law, Involution law, DeMorgan's Law and their applications;
Obtaining Sum of Product (SOP) and Product of Sum (POS) form from the Truth Table, Reducing Boolean Expression (SOP and POS) to its minimal form, Use of Karnaugh Map for minimization of Boolean expressions (up to 4 variables);
Basic Logic Gates (NOT, AND, OR, NAND, NOR) and their use in circuits.

UNIT 5: COMMUNICATION AND NETWORK CONCEPTS

Evolution of Networking: ARPANET, Internet, Interspace;
Different ways of sending data across the network with reference to switching techniques;
Data Communication terminologies: Concept of Channel, Baud, Bandwidth (Hz, KHz, MHz) and Data transfer rate (bps, kbps, Mbps, Gbps, Tbps);
Transmission media: Twisted pair cable, coaxial cable, optical fiber, infrared, radio link, microwave link and satellite link.
Network devices: Modem, RJ45 connector, Ethernet Card, Hub, Switch, Gateway;
Different Topologies- Bus, Star, Tree; Concepts of LAN, WAN, MAN;
Protocol: TCP/IP, File Transfer Protocol (FTP), PPP, Level-Remote Login (Telnet), Internet, Wireless/Mobile Communication, GSM, CDMA, WLL, 3G, SMS, Voice mail, Application Electronic Mail, Chat, Video Conferencing;
Network Security Concepts: Cyber Law, Virus threats and prevention, Firewall, Cookies,Hacking;
WebPages; Hyper Text Markup Language (HTML), eXtensible Markup Language (XML); Hyper Text Transfer Protocol (HTTP); Domain Names; URL; Protocol Address; Website, Web browser, Web Servers; Web Hosting.

CLASS XII (INFORMATICS PRACTICES)

UNIT 1: BUSINESS COMPUTING

Introduction to Open Source based software:
Terminology: OSS, FLOSS, GNU, FSF, OSI, W3C.
Definitions: Open Source Software, Freeware, Shareware, Proprietary software, Localisation, UNICODE
Softwares : Linux, Mozilla web browser, Apache server, MySQL, Postgres, Pango, OpenOffice, Tomcat, PHP, Python
Websites: www.sourceforge.net, www.openrdf.org, www.opensource.org, www.linux.com, www.linuxindia.net, www.gnu.org.
General concepts, User interfaces (Front End), Underlying Database (Back End), Integration User Interface and Database;
More application areas of Databases:
Inventory control, Financial Accounting, Pay-Accounting System, Invoicing Management System, Personal Management System / HRD System, Fees Management system, Result Analysis System, Admission Management System, Income Tax Management System;
Advanced Program Development Methodology: System Development Life Cycle, Relational Database Concept, Relational Database, Management System, Data Models (Entity Relationship Model), Entity and Entity Set, Attributes (Single, Composite and Multi-Valued), Relationship One-to-One, One-to-Many and Many-to-Many), Entity Relationship Modeling Conventions, Communicating with an RDBMS using SQL, Relational Database Management System, SQL Statements, About programming language in SQL.
Data Dictionary, Data Warehousing, Data Mining, Meta Data;
Object Modeling: Introduction to object oriented modeling using Unified Modeling Language Concepts only).
Client Server Computing: Concept of Client Server Computing.

UNIT 2: PROGRAMMING: Visual Basic

Review of Class XI;
Programming Fundamentals
Modules: Modules in Visual Basic- Form Modules, Standard Modules, and Class Modules;
Procedures: Procedures (General, Event, Function, Property);
Control Structures:
Revision of Decision Structure – IF, IF-THEN-ELSE, Select Case;
Revision of Looping Structure- Do While…Loop, Do…Loop While, For…Next, For Each…Next;
Functions: Concept of Functions, Defining and Use of User Defined functions, function to perform calculations, Parameterized Functions;
Library Functions (System Functions)
String Function: Space( ), Str( ), Right( ), Left( ), Mid( ), InStr( ), Len( ), Ltrim( ), Rtrim( ), Ucase(), Lcase( ), String( );
Numeric Function: Sgn( ), Val( ), Int( );
Time-Related Function: Now( ), Time( ), Minute( ), Month( );
Miscellaneous Function: MsgBox( ), InputBox( );
Types of forms: Single Document Interface (SDI) and Multiple Document Interface (MDI);
MDI Applications: Creating MDI form and Child form, Arranging Child Forms;
Accessing database from ORACLE using ODBC or ADO or OLEDB to connect with database.
Data Control: Accessing Data with the Data Control, Using Data-Aware Controls, Using Data Control Properties – Database Name, Exclusive, Options, Read Only, Record Source, Data Control Methods – Refresh, UpdateControls, UpdateRecord; Bound Controls: Adding Bound Text and Bound Label Controls. Data-Bound list Boxes, Grids, and Sub-Forms
ADO (ActiveX Data Objects): Connection Object, Command Object, and RecordSet Object, Special ADO Properties – Connection String (using single table), Command Text, Command Types, Cursor Locations, Cursor Types, Lock Types, Mode Types.
ADO Data Control: Simple Data linking using ADO Data Control Methods, ADO Data Control Events.

UNIT 3: RELATIONAL DATABASE MANAGEMENT SYSTEM

Review of RDBMS from Class XI

Database Fundamentals

Concept of Database Transaction, Committing a Transaction, Concept of "All or None" in a Transaction, Network Protocols Required (TCP/IP) for Data Communication, Stored Procedures, Concept of Database Fragmentation and Distributed Databases.

PL/SQL (Programming Language in SQL)

Importance of Writing Procedures, Declaring Variables: About PL/SQL, PL/SQL Block Structure, Program Constructs, Use of Variables, Handling Variables in PL/SQL, Types of Variables, Declaration, Naming Rules, Assigning Values to Variables, Initialization, and Keywords, Scalar Data types, Base Scalar Data Types, Scalar Variable Declaration, %TYPE attribute: for variabledeclaration, Declaring Boolean Variables, PL/SQL Record Structure, Referencing Non-PL/SQL variables, DBMS_OUTPUT.PUT_LINE;

Writing Executable Statements: PL/SQL Block Syntax and Guidelines, SQL functions in Code, SQL Functions in PL/SQL, PL/SQL Functions, Data type Conversion, Nested Blocks and Variable Scope, Operators in PL/SQL, Using Bind Variables, Programming Guidelines,

Determining Variable Scope, SQL Statements in PL/SQL, Retrieving data in PL/SQL, Manipulating Data using PL/SQL, Inserting Data, Updating Data, Deleting Data, Naming Conventions, Commit and Rollback Statements, SQL Cursor, and Cursor Attributes;

Writing Control Structures: Controlling PL/SQL Flow of Execution, IF statements, IFTHENELSE Statement Execution Flow, IF-THEN-ELSIF Statement Execution Flow, Building Logical Conditions, Logic Tables, Boolean Conditions, Iterative Control: LOOP Statement, Basic Loop, FOR Loop, While Loop;

Creating Procedures: Overview of Procedures, Syntax for Creating Procedures, Developing Stored Procedures and its Advantages, Creating a Stored Procedure, Procedure Parameter Modes, Creating Procedures with Parameters, IN and OUT parameters and Usage, DEFAULT Option for Parameters, Removing Stored Procedures;

Writing Cursors: Introduction to Cursors (Implicit and Explicit), Explicit Cursor Functions, Controlling Explicit Cursors, Declaring, Opening and Closing the Cursor, Fetching data from the Cursor, Explicit Cursor Attributes (%ISOPEN, %NOTFOUND, %ROWCOUNT), controlling multiple fetches, Cursors and Records, Cursor FOR Loops, Cursor FOR Loops using Sub Queries.

Triggers: Types of Triggers: Row-Level Triggers, Statement Level Triggers, BEFORE and AFTER Triggers, INSTEAD of Triggers, Valid Trigger Type, Trigger Syntax, Combining Trigger Types, Enabling and Disabling Trigger, Replacing Trigger, Dropping a Trigger.

Development of Data Base Applications (Application Domain): Student database for school, Employee database for a company, Library Database for Library Student database management system for school, Employee database management system for a company, Library Database management system for Library, Railway Reservation System, Hotel Reservation, Inventory Control System;

CLASS XII (MULTIMEDIA AND WEB TECHNOLOGY)

UNIT 1: COMPUTER SYSTEM

Database Terminology: Data, Record/Tuple, Table, Database
Concept of Keys: Candidate Key, Primary Key, Alternate Key, and Foreign Key;
Database Tool: Using MS– Access, Creating and Saving Table, Defining Primary Key, Inserting and Deleting Column, Renaming Column, Inserting records, Deleting Records, Modifying Records, and Table Relationship.

UNIT 2: WEB TECHNOLOGIES

Communication and network concepts
Evolution of Networking: ARPANET, Internet, Interspace;
Different ways of sending data across the network with reference to switching techniques;
Data Communication terminologies: Concept of Channel, Baud, Bandwidth (Hz, KHz, MHz) and Data transfer rate (bps, kbps, Mbps, Gbps, Tbps);
Transmission media: Twisted pair cable, coaxial cable, optical fiber, infrared, radio link, microwave link and satellite link.
Network devices: Modem, RJ45 connector, Ethernet Card, Hub, Switch, Gateway;
Different Topologies- Bus, Star, Tree; Concepts of LAN, WAN, MAN;
Protocol: TCP/IP, File Transfer Protocol (FTP), PPP, Level-Remote Login (Telnet), Internet, Wireless/Mobile Communication, GSM, CDMA, WLL, 3G, SMS, Voice mail, Application Electronic Mail, Chat, Video Conferencing;
Network Security Concepts: Cyber Law, Firewall, Cookies, Hackers and Crackers;
Introduction to Open Source based software
Terminology: OSS, FLOSS, GNU, FSF, OSI, W3C
Definitions: Open Source Software, Freeware, Shareware, Proprietary software, Localisation, UNICODE
Softwares : Linux, Mozilla web browser, Apache server, MySQL, Postgres, Pango, Open Office, Tomcat, PHP, Python
Websites: www.sourceforge.net, www.openrdf.org, www.opensource.org, www.linux.com, www.linuxindia.net, www.gnu.org.
Multimedia Application*:* Education (use of CAI tool), Entertainment , Edutainment, Virtual Reality, Digital Libraries, Information Kiosks, Video on Demand, Web Pages Video phone,Video conferencing and Health care.

UNIT 3: WEB DEVELOPMENT

Review Of HTML/DHTML, VBScript covered in Class XI.
Installation and Managing WEB-Server: Internet Information Server (IIS) / Personal Web Server (PWS).
Active Server Pages (ASP): Concept of ASP, features of ASP, other equivalent tools – JSP, PHP;
Constants: String and Numeric;
Data types: Integer, Floating Point (Single, Double), String, Date, Boolean, Currency, Variant, Object;
Variables: Explicit and Implicit Declaration;
Operators:
Arithmetic: +, - (Unary and Binary), *, /, \(integer division) mod, ^;
Comparison: <, >, <=, >=, <>, =;
Logical: AND, OR, NOT, XOR, EQV, IMP;
String Operator: & ,or ,+ (for Concatenation);
Functions:
Conversion functions: Abs(), CBool(), CByte(), CInt(), CStr(), CSng(), CLng(), CDate();
String Manipulation Functions: UCase(), LCase(), Len(), Left(), Right(), Mid(), LTrim(), InStr(),RTrim(), LTrim();
Time & Date Functions: Date(), Day(), Hour(), Left(), Len(), Minute(), Month(), Monthname(),Now();
Arrays: Declaration and use of 1 dimensional arrays;
Controls: IF..THEN, IF..THEN..ELSE..END IF, IF..THEN.. ELSEIF..THEN.. END IF, SELECT..CASE..END SELECT, FOR..NEXT, FOR EACH.. NEXT, DO WHILE..LOOP, DO..LOOP WHILE, DO UNTIL . LOOP;
Procedures and Functions, Passing parameters/arguments;
Concept of object model structure (client to server and server to client);

Objects:

Properties, Methods, Events, Setting Object properties, Retrieving Object properties, calling objects/methods;

Types of Objects: Response, Request, Application, Session, Server, ASPError;

Response Object: Write Method, AddHeader, AppendToLog, BinaryWrite, Using Shortcuts <%=value/expr%>, Controlling information: Buffer, Flush Clear, End;

Request Object: Request Object Collection: QueryString, Form, ServerVariables, Cookies, Client Certificate;

Application : Contents, Lock, Unlock, Remove, RemoveAll;

ASP Components: AD Rotator, Content Rotator, Counter, Page Counter, Permission Checker;

Text Files: Open and Read content from a text file;

Elementary Database Concepts: Concept of Table/Relation, Relationship, Candidate Key, Primary Key, Alternate Key, Foreign Key, Connecting with Databases: Creation of DSN, using OLE DB. Working on Database: Inserting, Retrieving, Modifying/Updation of records from Tables in Databases using server objects (ADODB. Connection, ADODB. Recordset);

Server Variables: HTTP_User_Agent, REMOTE_ADDER, REMOTE_HOST, SERVER_NAME;

UNIT 4: MULTIMEDIA AND AUTHORING TOOLS

Movie File Formats: AVI, MPEG, SWF, MOV, DAT;

Movie Frames: Concept of Frame, Frame Buffer, and Frame Rate;

Authoring Tools; Making Animation, Embedding Audio/Video, and Embedding on the page;

Multimedia Authoring Using Macromedia Flash

Making of Simple Flash Movie, Setting Properties, Frame Rate, Dimensions, and Background Color;

Scene: Concept of Scene, Duplicate Scene, Add Scene, Delete Scene, and Navigating between Scenes;

Layers: Concept of Layer, Layer Properties, Layer Name, Show/Hide/Lock layers, Type Layer - Normal/Guide/Mask, Outline Color, Viewing Layer as outline, Layer Height, Adding/deleting a layer;

Frame: Concept of Frame;

Creating a Key Frame, Inserting Text Into the Frame, Inserting Graphical Elements into frame, Converting Text/Graphics to Symbol, Inserting Symbol into the Frame, Setting SymbolProperty (Graphics/Button/Movie), Inserting Blank Frame, Inserting Blank Key Frame,Inserting Key Frame into the Blank frame, Selecting all/Specific frames of a Layer,

Copying/Pasting selected Frames,

Special Effects: Motion Tweening, Shape Tweening, Color effect, Inserting Sound Layer;

Testing a Scene and Movie;

Import/Export (Movie/Sound and other multimedia objects)

Publishing: Publishing A Flash Movie; Changing publish Settings; Producing SWF(Flash Movie), HTML page, GIF image, JPEG Image (*.jpg), PNG Image, Windows Projector (*.exe), Macintosh Projector (*.hqx), Quick Time (*.mov), Real Player (*.smil);

Testing with Publish Preview

ICSE & ISC SYLLABUS

N.M
ICSE 2010

3

COMPUTER APPLICATIONS (86)

CLASS IX

There will be one written paper of two hours duration carrying 100 marks and Internal Assessment of 100 marks.

The paper will be divided into two sections A & B.

Section A (Compulsory – 40 marks) will consist of compulsory short answer questions covering the entire syllabus.

Section B (60 marks) will consist of questions which will require detailed answers and there will be a choice of questions in this section

THEORY – 100 Marks

1. Operating System

i) Command User Interface

The need for an Operating System, the Booting Process, Directory handling, Absolute and Relative path names, File handling, Disk handling Commands, Batch Files.

Students should know how to Boot a system and how to shut it down properly.

What are directories and their hierarchical structure, the concept of root, parent and child directories, Absolute and Relative path names.

File naming conventions. Wild card characters. Viewing and changing file attributes. Formatting and un-formatting a disk. Taking and restoring backups, scanning the disk for errors. Defragmenting the disk, Batch files, automatically executable batch files.

ii) Graphic User Interface

Working with the Graphical User Interface (GUI), Elements of a GUI, handling files and directories under GUI, managing the desktop.

Students should know how to start a GUI and how to shut it down properly.

Concept of an active window. The student should be familiar with the icons, the buttons, the bars and the menus in a window. It is not necessary to know every option in each menu.

Resizing a window. Handling multiple windows.

Creating folders. Copying items. Moving items. Deleting Items. Setting attributes of items.

Creating shortcuts on the desktop, arranging the folders on the desktop.

iii) Installing the Operating System

Installing the Operating System and other software, setting up printers.

The teacher should demonstrate the following:

Installation of an Operating System, settings related to connected peripherals.

Installing and uninstalling other software. Installing and uninstalling printers.

2. The Internet

Communicating through Computers, creating an e-mail ID, e-mail and ftp, POP and IMAP and SMTP for mail and the ftp protocol for file transfer, using the World Wide Web, HTTP and web servers, downloading information, using Search engines, configuring the Internet.

Students should know how to connect to the Internet, use a browser, make an e-mail ID, send and receive email.

Students should know what is a protocol and why is it needed.

Downloading information, use of search engines to perform a simple search and Boolean operators to fine tune a search.

The teacher should demonstrate how to configure the Internet and Install a Modem.

3. Computing and Ethics

Ethical issues in computing; intellectual property rights; protection of individual's right to privacy; data protection on the internet; protection against Spam; software piracy, cyber crime, hacking, protection against malicious intent and malicious code.

.eresting discussions can be organized in the class .egarding ethical issues. Students can gather more information from the internet. The stress should be on following good etiquette and ethical practices.

4. Word Processing

Starting and closing the word processor under a graphic user interface; creating and saving a document; protecting a document; editing a document, find and replace, page layout, printing a document, formatting a document, enhancing a document, proofing a document, mail merge; labels & envelopes; creating indexes and table of contents. Templates, embedding objects.

Students should know how to start and close a word processor under the graphic user interface. They should know how to create a document, save it in an appropriate area and protect it.

Copying, moving, deleting text and reviewing recent changes. Finding and replacing text- with all possible options.

Page layout, size, orientation, margins, and print preview, printing all the pages or selections.

Changing the font and font size, applying attributes – bold, italics, underline. Justification, word wrap, line spacing.

Headers and footers, borders, bulleting and numbering, indenting.

Spell check, grammar check, using a thesaurus.

Basic concepts of mail merge, master document, data source. Merging the document and printing it, printing labels and envelopes.

Students should know how to create indexes and table of contents. They should be able to create templates and use readymade templates to create documents. They should be able to embed objects in the document.

5. Multimedia Presentation

The art and science of communication. Planning a presentation; creating and saving a presentation in different formats; slide layout and transition; working with graphics, embedding audio and video clips; recording dialogue and sounds; embedding scanned images and objects from other software. Integrating the presentation with word processors and the internet, animation and design enhancement.

Criteria to be kept in mind includes: the target audience, the core message, visualization, aesthetics, medium, form and style. Students should plan each slide for the presentation before they actually start working on a computer. They should pay special attention to the content, originality, layout, graphics, text, correct grammar, colour and links. The text, graphics, background and foreground hues, music, dialogue and animation should harmonize.

The students should be able to create a presentation and save it in various formats.

The layout and transition of slides, animation and design should be appropriate for the topic. Students may create their own graphics or insert pictures/scanned images from other sources. They may record their own voices, music and video footage or may use audio/video clips from other sources. Students should be able to import material from word processors, spreadsheets, databases and internet or build links to these.

Important: While using material from other sources, students should take special care not to violate any copyright laws.

6. Spreadsheets

The need for spreadsheets; creating and saving a spreadsheet; working with formulae; relative and absolute referencing; editing a spreadsheet, using functions, working with ranges, graphs and charts; formatting a spreadsheet; working with multiple spreadsheets; printing spreadsheets; importing external data and working with it. Tools for analysis.

The need for spreadsheets for data manipulation, analysis and future projections.

The electronic spreadsheet as an array of rows and columns, cell, cell address, cell pointer.

Creating and saving a spreadsheet.

Entering different types of data, generating series, entering and copying formulae, difference between absolute and relative references.

Copying, moving, inserting and deleting entries in cells/rows and columns. Automatic recalculation.

Mathematical, statistical, text, date and time, logical and look up functions as required by the students.

Selecting, copying, moving, deleting and naming ranges.

Creating charts/graphs of different types depending upon the requirement of the problem.

Formatting numbers, dates and text.

Linking multiple spreadsheets, copying and moving entries from one sheet to another, saving a group of spreadsheets.

Print preview, printing a range, printing the whole worksheet, printing with/without gridlines.

Importing data from word processors, internet and databases and manipulating it according to the need of the problem.

What if analysis, scenario building and future projections depending upon the need of the problem and tools available in the package.

7. Database Packages

The need for database management; creating and saving a database; editing a database; performing calculations; modifying the structure of a database; sorting, indexing; querying; mailing labels; report generation. Working with multiple databases, object linking and embedding, creating applications.

The need for database management for handling vast amount of data- storing, sorting, summarizing, classifying and retrieving quickly.

Defining the structure of a database, entering data of various types, saving it in an appropriate area.

Adding, deleting and modifying records, global editing.

Performing calculations on one record or a group of records.

Modifying the structure of a database by inserting, deleting or modifying fields.

Sorting on one field/ multiple fields, sorting selected records/ all the records.

Indexing on one field/ multiple fields. The need for re- indexing. Sorting vs. Indexing.

Setting query condition, Relational and Logical Operators, setting query using multiple conditions.

Creating mailing labels, defining size, layout, content. Preview and printing of labels.

Generating detailed or summary reports.

Setting relations between databases, setting queries on multiple databases.

Linking objects/embedding objects, linking vs. embedding.

Creating database applications depending upon the requirement of the user.

8. Elementary Concept of Objects and Classes

Modelling entities and their behaviour by objects; a class as a specification for objects and as an object factory; computation as message passing/function calls between objects (many examples should be done to illustrate this). Objects encapsulate state (attributes) and have behaviour (functions). Class as a user defined type.

Show how various entities in the real world can be thought of as a bunch of attributes and different operations which can be performed on this bunch of attributes. If we bundle the bunch of attributes and let this bunch behave according to the operations that can be done we have an *object*. The current values of the attributes define the state of the object and the operations define the behaviour. Operations are often referred to as member functions, methods or messages. Students should understand that they mean the same thing. Sometimes it is more intuitive to talk in terms of one object sending a message to another object.

A class may be regarded as a blueprint to create objects. It may be viewed as a factory that produces similar objects. A class may also be considered as a new data type created by the user, that has its own functionality.

Possible examples: a calculator, a student, a number (most number problems can be defined as operations on number objects). Analyze each object and show how each contains attributes and responds to certain messages or permits certain operations.

Emphasize that an object is an *instance* of a class. A single object is just a bundle of values, one for each attribute in the class.

9. Values and types

Primitive types (like int, float, boolean, etc.) and their representation and ranges, operations on primitive values, expressions, assignment (assignment is also an expression), attributes, constructors, functions. Return values and function parameters, passing parameters by value.

Introduce the primitive types and the range of values each represents. Discuss all the operations that can be done with primitive types. This is a fairly long list but it needs to be done thoroughly since it greatly increases the range of problems that a student can attempt. Discuss precedence and associativity of operators. While discussing *boolean* values do truth tables for the basic logical connectives and show the duality between *and* & *or* (DeMorgan's theorem) - students often have difficulty with boolean expressions.

Constructors are used to construct objects. Show how arguments can be passed to a constructor to create a stable object.

Start with simple *get, set* functions to motivate functions as the definition of behaviour for an object. Introduce *System.out.println* and *System.out.print,* they are very useful for simple output. In BlueJ all variables of class type are references to objects. Objects are always created in dynamic memory or the heap. All arguments are always passed by value. One way to think about it is to assume that all functions always have an implicit argument- the object on which the invocation is made (referred to by *this*). Also discuss the return type. Discuss the use of *void.*

Show how types can catch errors during compilation and so prevent errors when the program executes.

10. Conditionals and non-nested loops

if-then, if-then-else.

Fixed number of iterations- the for loop. Unknown number of iterations - while loop, do-while loop.

The conditional constructs are easy to motivate.

Loops are fundamental to computation and their need should be shown by examples. Select examples requiring both pre-determined and undetermined number of iterations. *for* loops should normally be used for iterations which have a pre-determined number of iterations. And *while-do, do-while* when the number of iterations are not known a priori. Explain the semantics of each loop very carefully - students often forget what the condition being true/false actually implies.

Examples: various number based problems: prime numbers, composite numbers, perfect numbers, fibonacci numbers, etc., min-max.

CLASS X

There will be one written paper of two hours duration carrying 100 marks and Internal Assessment of 100 marks.

The paper will be divided into two sections A & B.

***Section A** (Compulsory – 40 marks) will consist of compulsory short answer questions covering the entire syllabus.*

***Section B** (60 marks) will consist of questions which will require detailed answers and there will be a choice of questions in this section*

THEORY – 100 Marks

1. Revision of Class IX Syllabus

(i) Elementary Concept of Objects and Classes.

(ii) Values and types.

(iii) Conditionals and non-nested loops.

2. Class as the Basis of all Computation

Objects encapsulate state and behaviour – numerous examples; member variables; attributes or features. Variables define state; member functions; Operations/methods/ messages/ functions define behaviour.

Classes as abstractions for sets of objects; class as an object factory; concept of type, primitive data types, composite data types. Variable declarations for both types; difference between the two types. Objects as instances of a class. modelling by composition.

It has already been shown to the students that the current values of the attributes define the state of the object and the operations define the behaviour.

Using the above, explain how data comes in various forms. Some of it is primitive - numbers of various kinds, characters; other data is composite - a string, an object also represents composite data (consisting of its attributes).

Possible examples: a student, a queue, an oven, a bank, a number (most number problems can be defined as operations on number objects), many more are possible. Analyze each object and show how each contains attributes and responds to certain messages or permits certain operations.

A class is an abstraction for a set of objects. It describes the attributes that each object will have and the behavior it can exhibit through the functions defined in the class. Alternately, it can be thought of as a factory that can produce objects using constructors (see 3 below). Emphasize that an object is an *instance* of a class. A single object is just a bundle of values, one for each attribute in the class. Emphasize that attribute values need not be primitive but can be themselves objects - for example, the birth date of a student will be a Date object. Discuss the primitive types and the range of values each represents. Show how each attribute and function needs the specification of the type of value it represents or returns. Also, show how types can catch errors during compilation and so prevent errors when the program executes.

3. Constructors

Default constructor, constructors with arguments, default initialization. Overloading constructors.

Students already know that constructors are used to construct objects.

Show how if no constructor is defined a default constructor without any arguments is always defined. Multiple constructors are possible, if they differ in their arguments. This introduces the concept of overloading. Using the earlier examples, define different constructors for each kind of class. Discuss the details of default initialization, and the *null* value.

4. Functions

Function as a way to define operations/methods/ messages; pure functions return values and do not change state; impure functions may return values but also change state, return types, arguments to functions, function prototype and function signature, overloading. Variables of class type as references to an object, invocation of functions on objects through the reference, the concept of this. Argument passing in functions, pass by value, what happens when a reference is passed - side effects.

Students have already been familiarized with functions. Sometimes complex behaviour may have to be implemented using more than one function (this can be used to motivate *private* functions). Try to use the same examples that were used earlier. *get* functions are pure functions (provided they do not change the value of any variable - typically they do not) because the state of the object is not changed. On the other hand, some functions (ex. *set*) change the state of the object in question and are therefore called *impure* functions or procedures. Show the need for overloading functions. Introduce the *void* type that signals that a function is impure. In BlueJ all variables of class type are references to objects. Objects are always created in dynamic memory or the heap. All arguments are always passed by value. However, since variables referring to objects are references, any change inside a function will be reflected outside (that is there is a side effect). But this is not true for variables of primitive type. Discuss invocation of functions on objects (through the reference). Discuss the concept of *this* with a reference to the object on which the invocation is made again. A good way to think about it is to assume that all functions always have an implicit argument - the object on which the invocation is made (referred to by *this*).

5. Class as a User Defined Type

Class as a composite type, distinction between primitive type and composite or class types.

Students know that a class may be considered as a new data type created by the user, that has its own functionality.

The distinction between primitive and composite types should be discussed through examples. Show how classes allow user defined types in programs. All primitive types have corresponding class wrappers. Do some example wrappers - *Integer*, *Boolean*.

6. Decision Making

Application of if-then, if-then-else, switch (default, break).

The conditional and decision making constructs are easy to teach. The ?: operator should be introduced at this point.

7. Iterations.

Utilization of loops. Fixed number of iterations. The for loop, Unknown number of iterations - while loop, do-while loop, (for loop can also be used but should be used mainly when number of iterations are fixed), continue, break. Nested loops.

Loops are fundamental to computation and have been done earlier. Show how each kind of loop can be converted to the other form of the loop. So though only one is enough, it is best to have all of them since some loops are better expressed using one construct rather than another - give examples of how some constructs are best expressed using a particular kind of loop. Introduce nested loops.

8. Using Library Classes

Simple input, output. String, static variables and static methods, packages and import statements.

It is important to avoid discussing the details of libraries. Teach the student how to browse the documentation for classes in the libraries and illustrate their use. The *String* class is very appropriate since students already have good intuition about strings. BlueJ input requires a little extra work so it is best to define a class which allows input of different types of primitive values. The student can then use this class and its functions to do reading fairly easily. This is also the right time to introduce the concept of a package that allows a user to organize sets of related classes implementing a particular functionality. This leads naturally to the *import* statement, which allows unqualified access to the classes within a package.

9. Encapsulation

Private, public, scope and visibility rules; packages and package level access.

Motivate need for separating implementation from the interface. Visibility rules for *private*, package and *public* access specifiers. Scope of variables, instance variables, argument variables, local variables, holes in the scope.

10. Arrays

Arrays and their usage; sorting algorithms - selection sort and bubble sort; Search in sorted arrays. The class Object - compatible with all class types.

Example of a composite type. Array creation. Sorting and searching algorithms should be discussed. Point out how it is difficult to sort arbitrary objects. One way out is by passing your own comparator objects. Need for a class with which all other classes are compatible. The class *Object*. Since inheritance will be done later, type compatibility has to be explained simply as either the same type or the formal type is *Object* and the actual type is some class type. Type checking for assignment, and when actual parameter types are matched with formal parameter types during a function call, should be explained.

11. Operations on Files

Streams - byte and character streams, files and operations on files, tokens and String Tokenizer and Stream Tokenizer classes.

<u>Important:</u> The teachers may write classes for I/O and give them to the students to use. The students may need Operations on Files for their projects. ***However no questions shall be asked in the theory paper from Operations on Files.***

BlueJ I/O is stream based and due to historical reasons was byte oriented initially. Now character oriented I/O is also present as well as classes to convert byte streams to character streams. They can see the code for the class provided earlier for doing input. The class *File* should be discussed. The concept of a token should be discussed and the use of *StringTokenizer* and *StreamTokenizer* classes for extracting tokens should be explained.

N.M

ISC 2010

COMPUTER SCIENCE (868)

CLASS XI

There will be two papers in the subject:

Paper I: Theory - 3 hours ....100 marks

Paper II: Practical - 3 hours ....100 marks

PAPER I -THEORY

Paper 1 shall be of 3 hours duration and be divided into two parts.

__Part I (30 marks):__ This part will consist of compulsory short answer questions, testing knowledge, application and skills relating to the entire syllabus.

*__Part II (70 marks):__ This part will be divided into three Sections, A, B and C. Candidates are required to answer **three** questions out of **four** from Section A and **two** questions out of **three** in each of the Sections B and C. Each question in this part shall carry 10 marks.*

SECTION A

Basic Computer Hardware and Software

1. *Numbers*

Representation of numbers in different bases and interconversion between them (e.g. binary, octal, decimal, hexadecimal). Addition and subtraction operations for numbers in different bases.

Introduce the positional system of representing numbers and the concept of a base. Discuss the conversion of representations between different bases using English or pseudo code. These algorithms are also good examples for defining different functions in a class modelling numbers (when programming is discussed). For addition and subtraction use the analogy with decimal numbers, emphasize how carry works (this will be useful later when binary adders are discussed).

2. *Encodings*

(a) *Binary encodings for integers and real numbers using a finite number of bits (sign-magnitude, twos complement, mantissa-exponent notation). Basic operations on integers and floating point numbers. Limitations of finite representations.*

Signed, unsigned numbers, least and most significant bits. Sign-magnitude representation and its shortcomings (two representations for 0, addition requires extra step); twos-complement representation. Operations (arithmetic, logical, shift), discuss the basic algorithms used for the arithmetic operations. Floating point representation: normalized scientific notation, mantissa-exponent representation, binary point (discuss trade-off between size of mantissa and exponent). Single and double precision. Arithmetic operations with floating point numbers. Properties of finite representation: overflow, underflow, lack of associativity (demonstrate this through actual programs).

(b) *Characters and their encodings (e.g. ASCII, Unicode).*

Discuss the limitations of the ASCII code in representing characters of other languages. Discuss the Unicode representation for the local language. Java uses Unicode, so strings in the local language can be used (they can be displayed if fonts are available) – a simple table lookup for local language equivalents for Latin (i.e. English) character strings may be done. More details on Unicode are available at www.unicode.org.

,h level structure of computer

Block diagram of a computer system with details of (i) function of each block and (ii) interconnectivity and data and control flow between the various blocks.

Develop the diagram by successive refinement of blocks till all the following have been covered: ALU, RAM, cache, the buses (modern computers have multiple buses), disk (disk controller and what it does), input/output ports (serial, parallel, USB, network, modem, line-in, line-out etc.), devices that can be attached to these ports (e.g keyboard, mouse, monitor, CDROM, DVD, audio input/output devices, printer, etc.). Clearly describe the connectivity and the flow of data and control signals.

r. Basic architecture of typical simple processor and its assembly language

(a) Basic architecture of the 8085 microprocessor. Instruction set, addresses, addressing modes, simple machine language programs using the different addressing modes, execution of machine language programs, input and output.

The idea here is to discuss a concrete microprocessor instead of an abstract computer thus giving students a clearer understanding of how a typical computer works. The aim is not to know all details of the 8085 microprocessor. The basic features that must be covered are: (i) Structure of memory; (ii) registers - A-register (accumulator), general (B, C, D, E, H, L in 8-bit individual and paired 16-bit modes), program counter, stack pointer, flag; (iii) addressing modes (immediate, direct, register, register-indirect); (iv) Instruction set (data transfer, arithmetic, logical, conditional and transfer of control, input/output). Interrupts are **not** included. Since many free simulators are available, the students should actually write, run and observe what happens when a machine language program runs. Example, machine and assembly language programs: evaluating simple expressions, adding a sequence of numbers, finding the minimum and/or maximum of a sequence of numbers, using finding the minimum / maximum to do sorting of a sequence of numbers. In particular, discuss how the stack can be used for calling and returning from subprograms. Emphasize how data and program look alike and depend on the interpretation used.

(b) Assembly language of 8085, simple assembly language programs, assembly process and assembler.

Discussion of the assembly language should be done along with the instruction set (previous section). Emphasize how it is easier to program in assembly language than in machine language. Assembly process: symbol table and its use in translating a program to machine language. Macros are not included.

5. Propositional logic, hardware implementation, arithmetic operations

(a) Propositional logic, well formed formulae, truth values and interpretation of well formed formulae, truth tables.

Propositional variables; the common logical connectives (~ (not), ∧ (and), ∨ (or), ⇒ (implication), ⇔ (equivalence)); definition of a well-formed formula (wff); representation of simple word problems as wff (this can be used for motivation); the values **true** and **false**; interpretation of a wff; truth tables; satisfiable, unsatisfiable and valid formulae.

(b) Logic and hardware, basic gates (AND, NOT, OR) and their universality, other gates (NAND, NOR, XOR); inverter, half adder, full adder.

Show how the logic in (a) above can be realized in hardware in the form of gates. These gates can then be combined to implement the basic operations for arithmetic. Tie up with the arithmetic operations on integers discussed earlier in 2 (a).

Memory

(a) Memory - construction of a memory bit using a flip-flop, D-flip-flop and its use in constructing registers.

(b) Memory organization and access; parity; memory hierarchy - cache, primary memory, secondary memory.

The access time differences between the different kinds of memory; size differences; locality of reference and cache memory.

System and other software

Boot process. Operating system as resource manager, command processing, files, directories and file system. Commonly available programs (editors, compilers, interpreters, word processors, spread sheets etc.).

Boot process step-by-step from power on till the prompt. In OS discuss: (i) all the resources (processor, memory, i/o) that need to be managed in a computer; (ii) what is meant by managing these resources. Logical structure of data storage on disk using logical disks, hierarchical directories and files. Distinguish between interpreters and compilers. In particular discuss the *javac* and *java* programs.

SECTION B

e programming element in the syllabus is aimed at algorithmic problem solving and **not** *merely rote learning of Java syntax. The Java version used should 1.5 or later. For programming, the students can use any text editor and the javac and java programs or any development environment: for example, BlueJ, lipse, NetBeans etc. BlueJ is strongly recommended its simplicity, ease of use and because it is very well suited for an 'objects first' approach.*

Introduction to algorithmic problem solving using Java

Note that topics 9 to 13 will get introduced almost simultaneously when classes and their definitions are introduced.

9. Objects

(a) Objects as data (attributes) + behaviour (methods or functions); object as an instance of a class. Constructors.

Difference between object and class should be made very clear. BlueJ (www.bluej.org) and Greenfoot (www.greenfoot.org) can be used for this purpose. Constructor as a special kind of function; the *new* operator; multiple constructors with different argument structures; constructor returns a reference to the object.

(b) Analysis of some real world programming examples in terms of objects and classes.

Use simple examples like a calculator, date, number etc. to illustrate how they can be treated as objects that behave in certain well-defined ways and how the interface provides a way to access behaviour. Illustrate behaviour changes by adding new functions, deleting old functions or modifying existing functions.

10. Primitive values, wrapper classes, types and casting

Primitive values and types: int, short, long, float, double, boolean, char. Corresponding wrapper classes for each primitive type. Class as type of the object. Class as mechanism for user defined types. Changing types through user defined casting and automatic type coercion for some primitive types.

Ideally, everything should be a class; primitive types are defined for efficiency reasons; each primitive type has a corresponding wrapper class. Classes as user defined types. In some cases types are changed by automatic coercion or casting – e.g. mixed type expressions. However, casting in general is not a good idea and should be avoided, if possible.

Variables, expressions

Variables as names for values; expressions (arithmetic and logical) and their evaluation (operators, associativity, precedence). Assignment operation; difference between left hand side and right hand side of assignment.

Variables denote values; variables are already defined as attributes in classes; variables have types that constrain the values it can denote. Difference between variables denoting primitive values and object values – variables denoting objects are references to those objects. The assignment operator = is special. The variable on the lhs of = denotes the memory location while the same variable on the rhs denotes the contents of the location e.g. i=i+2.

Statements, scope

Statements; conditional (if, if-then-else, switch-break, ?: ternary operator), looping (for, while-do, do-while, continue, break); grouping statements in blocks, scope and visibility of variables.

Describe the semantics of the conditional and looping statements in detail. Evaluation of the condition in conditional statements (esp. difference between || and | and && and &). Emphasize fall through in *switch* statement. Many small examples should be done to illustrate control structures. Printing different kinds of patterns for looping is instructive. When number of iterations are known in advance use the *for* loop otherwise the *while-do* or *do-while* loop. Express one loop construct using the others. For e.g.:

for (<init>; <test>; <inc>) <stmt>; is equivalent to:

(i) Using while

<init>; while <test> {<stmt>; <inc> }

(ii) Using do-while

<init>; if !<test> do <stmt>; <inc> while <test>;

Nesting of blocks. Variables with block scope, function scope, class scope. Visibility rules when variables with the same name are defined in different scopes.

13. Functions

Functions/methods (as abstractions for complex user defined operations on objects), functions as mechanisms for side effects; formal arguments and actual arguments in functions; different behaviour of primitive and object arguments. Static functions and variables. The ***this*** *variable. Examples of algorithmic problem solving using functions (various number theoretic problems, finding roots of algebraic equations).*

Functions are like complex operations where the object is implicitly the first argument. Variable **this** denotes the current object. Functions typically return values, they may also cause side-effects (e.g. change attribute values of objects) – typically functions that are only supposed to cause side-effects return *void* (e.g. *Set* functions). Java passes argument by value. Illustrate the difference between primitive values and object values as arguments (changes made inside functions persist after the call for object values). Static definitions as class variables and class functions visible and shared by all instances. Need for static functions and variables. Introduce the *main* method – needed to begin execution.

14. Arrays, strings

(a) *Structured data types – arrays (single and multi-dimensional), strings. Example algorithms that use structured data types (e.g. searching, finding maximum/minimum, sorting, solving systems of linear equations, substring, concatenation, length, access to char in string, etc.).*

Storing many data elements of the same type requires structured data types – like arrays. Access in arrays is constant time and does not depend on the number of elements. Structured data types can be defined by classes – *String*. Introduce the Java library *String* class and the basic operations on strings (accessing individual characters, various substring operations, concatenation, replacement, index of operations).

(b) *Basic concept of a virtual machine; Java virtual machine; compilation and execution of Java programs (the javac and java programs).*

The JVM is a machine but built as a program and not through hardware. Therefore it is called a virtual machine. To run, JVM machine language programs require an interpreter (the *java* program). The advantage is that such JVM machine language programs (.class files) are portable and can run on any machine that has the *java* program.

(c) *Compile time and run time errors; basic concept of an exception, the Exception class, catch and throw.*

Differentiate between compile time and run time errors. Run time errors crash the program. Recovery is possible by the use of exceptions. Explain how an exception object is created and passed up until a matching *catch* is found. This behaviour is different from the one where a value is returned by a deeply nested function call. It is enough to discuss the *Exception* class. Sub-classes of *Exception* can be discussed after inheritance has been done in Class XII.

SECTION C

13. *Elementary data structures and associated algorithms, basic input/ouput*

(a) *Class as a contract; separating implementation from interface; encapsulation; private and public.*

Class is the basic reusable unit. Its function prototypes (i.e. the interface) work as a visible contract with the outside world since others will use these functions in their programs. This leads to encapsulation (i.e. hiding implementation information) which in turn leads to the use of private and public for realizing encapsulation.

(b) *Interfaces in Java; implementing interfaces through a class; interfaces for user defined implementation of behaviour.*

Motivation for interface: often when creating reusable classes some parts of the exact implementation can only be provided by the final end user. For example in a class that sorts records of different types the exact comparison operation can only be provided by the end user. Since only he/she knows which field(s) will be used for doing the comparison and whether sorting should be in ascending or descending order be given by the user of the class.

Emphasize the difference between the Java language construct *interface* and the word interface often used to describe the set of function prototypes of a class.

(c) *Basic data structures (stack, queue, dequeue): implementation directly through classes; definition through an interface and multiple implementations by implementing the interface. Basic algorithms using the above data structures.*

A data structure is a data collection with well defined operations and behaviour or properties. The behaviour or properties can usually be expressed formally using equations or some kind of logical formulae. Consider for e.g. a stack with operations defined as follows:

void push(Object o)

Object pop()

boolean isEmpty()

Object top()

Then, for example the LIFO property can be expressed by (assume s is a stack):

if *s.push(o); o1=pop()* then *o* ≡ *o1*

What the rule says is: if *o* is pushed on the stack *s* and then it is popped and *o1* is the object obtained then *o*, *o1* are identical.

Another useful property is:

if *s.isEmpty() == true* then *s.pop() = ERROR*

It says that popping an empty stack gives ERROR.

Similarly, several other properties can also be specified. It is important to emphasize the behavioural rules or properties of a data structure since any implementation must guarantee that the rules hold.

Some simple algorithms that use the data structures:

i) For stack: parentheses matching, tower of Hanoi, nested function calls; solving a maze.

ii) For queue: scheduling processes, printers, jobs in a machine shop.

d) Basic input/output using Scanner and Printer classes from JDK; files and their representation using the File class, file input/output; input/output exceptions. Tokens in an input stream, concept of whitespace, extracting tokens from an input stream (StringTokenizer class).

The *Scanner* class can be used for input of various types of data (e.g. int, float, char etc.) from the standard input stream or a file input stream. The *File* class is used model file objects in the underlying system in an OS independent manner. Similarly, the *Printer* class handles output. Only basic input and output using these classes should be covered.

Discuss the concept of a token (a delimited continuous stream of characters that is meaningful in the application program – e.g. words in a sentence where the delimiter is the blank character). This naturally leads to the idea of delimiters and in particular whitespace and user defined characters as delimiters. As an example show how the *StringTokenizer* class allows one to extract a sequence of tokens from a string with user defined delimiters.

(e) Concept of recursion, simple recursive functions (e.g. factorial, GCD, binary search, conversion of representations of numbers between different bases).

Many problems can be solved very elegantly by observing that the solution can be composed of solutions to 'smaller' versions of the same problem with the base version having a known simple solution. Recursion can be initially motivated by using recursive equations to define certain functions. These definitions are fairly obvious and are easy to understand. The definitions can be directly converted to a program. Emphasize that any recursion must have a base case. Otherwise, the computation can go into an infinite loop. Illustrate this by removing the base case and running the program. Examples:

(i) Definition of factorial:

factorial(0) = 1 //base case

*factorial(n) = n * factorial(n-1)*

(ii) Definition of GCD:

gcd(m, n) =

if (m==n) then n //base case

else if (m>n) then gcd(m-n, n)

else gcd(m, n-m)

(iii) Definition of Fibonacci numbers:

fib(0) = 1 //base case

fib(1) = 1 //base case

fib(n) = fib(n-1)+ fib(n-2)

The tower of Hanoi is a very good example of how recursion gives a very simple and elegant solution where as non-recursive solutions are quite complex. Discuss the use of a stack to keep track of function calls. The stack can also be used to solve the tower of Hanoi problem non-recursively.

(f) Concrete computational complexity; concept of input size; estimating complexity in terms of functions; importance of dominant term; best, average and worst case.

Points to be given particular emphasis:

(i) Algorithms are usually compared along two dimensions – amount of space (that is memory) used and the time taken. Of the two the time taken is usually considered the more important. The motivation to study time complexity is to compare different algorithms and use the one that is the most efficient in a particular situation.

(ii) Actual run time on a particular computer is not a good basis for comparison since it depends heavily on the speed of the computer, the total amount of RAM in the computer, the OS running on the system and the quality of the compiler used. So we need a more abstract way to compare the time complexity of algorithms.

(iii) This is done by trying to approximate the number of operations done by each algorithm as a function of the size of the input. In most programs the loops are important in deciding the complexity. For example in bubble sort there are two nested loops and in the worst case the time taken will be proportional to n(n-1) where n is the number of elements to be sorted.

Similarly, in linear search in the worst case the target has to be compared with all the elements so time taken will be proportional to n where n is the number of elements in the search set.

(iv) In most algorithms the actual complexity for a particular input can vary. For example in search the number of comparisons can vary from 1 to n. This means we need to study the best, worst and average cases. Comparisons are usually made taking the worst case. Average cases are harder to estimate since it depends on how the data is distributed. For example in search, if the elements are uniformly distributed it will take on the average n/2 comparisons when the average is taken over a statistically significant number of instances.

(v) Comparisons are normally made for large values of the input size. This means that the dominant term in the function is the important term. For example if we are looking at bubble sort and see that time taken can be estimated as: $a*n^2 + b*n + c$ where n is the number of elements to be sorted and a, b, c are constants then for large n the dominant term is clearly n^2 and we can in effect ignore the other two terms.

Implementation of algorithms to solve problems

The students are required to do lab assignments in the computer lab concurrently with the lectures. Programming assignments should be done such that each major topic is covered in at least one assignment. Assignment problems should be designed so that they are non-trivial and make the student do algorithm design, address correctness issues, implement and execute the algorithm in Java and debug where necessary.

Self explanatory.

Social context of computing and ethical issues

(a) Intellectual property and corresponding laws and rights, software as intellectual property.

(b) Software copyright and patents and the difference between the two; trademarks; software licensing and piracy.

(c) Free software foundation and its position on software, open source software, various types of licensing (e.g. GPL, BSD).

(d) Privacy, email etiquette, spam, security issues, phising.

Social impact and ethical issues should be discussed and debated in class. The important thing is for students to realise that these are complex issues and there are multiple points of view on many of them and there is no single 'correct' or 'right' view.

PAPER II - PRACTICAL

This paper of three hours duration will be evaluated internally by the school.

The paper shall consist of three programming problems from which a candidate has to attempt any one. The practical consists of the two parts:

(1) Planning Session

(2) Examination Session

The total time to be spent on the Planning session and the Examination session is three hours. After completing the Planning session the candidates may begin with the Examination session. A maximum of 90 minutes is permitted for the Planning session. However, if the candidates finish earlier, they are to be permitted to begin with the Examination session.

Planning Session

The candidates will be required to prepare an algorithm and a hand written Java program to solve the problem.

Examination Session

The program handed in at the end of the Planning session shall be returned to the candidates. The candidates will be required to key-in and execute the Java program on seen and unseen inputs individually on the Computer and show execution to the examiner. A printout of the program listing, including output results should be attached to the answer script containing the algorithm and handwritten program. This should be returned to the examiner. The program should be sufficiently documented so that the algorithm, representation and development process is clear from reading the program. Large differences between the planned program and the printout will result in loss of marks.

rs should maintain a record of all the gnments done as part of the practical work ough the year and give it due credit at the time of nulative evaluation at the end of the year. Students are expected to do a ***minimum*** *of twenty assignments the year.*

rks (out of a total of 100) should be distributed as given below:

ntinuous Evaluation

Candidates will be required to submit a work file taining the practical work related to programming ignments done during the year.

Programming assignments done throughout the year ernal evaluation) *- 20 marks*

Terminal Evaluation

Solution to programming problem on the computer *- 60 marks*

(Marks should be given for choice of algorithm and implementation strategy, documentation, correct output on known inputs mentioned in the question paper, correct output for unknown inputs available only to the examiner.)

Viva-voce *- 20 marks*

(Viva-voce includes questions on the following aspects of the problem attempted by the student: the algorithm and implementation strategy, documentation, correctness, alternative algorithms or implementations. Questions should be confined largely to the problem the student has attempted).

CLASS XII

There will be two papers in the subject:

per I: Theory-	*3 hours*	*...100 marks*
Paper II: Practical-	*3 hours*	*...100 marks*

PAPER I-THEORY

per 1 shall be of 3 hours duration and be divided into two parts.

Part I (30 marks): *This part will consist of mpulsory short answer questions, testing knowledge, application and skills relating to the entire labus.*

Part II (70 marks): *This part will be divided into three Sections, A, B and C. Candidates are required to wer* ***three*** *questions out of* ***four*** *from Section A and two questions out of* ***three*** *in each of the Sections B and C. Each question in this part shall carry 10 rks.*

SECTION A

Boolean Algebra

(a) Propositional logic, well formed formulae, truth values and interpretation of well formed formulae (wff), truth tables, satisfiable, unsatisfiable and valid formulae. Equivalence laws and their use in simplifying wffs.

Propositional variables; the common logical connectives (~ (not), $\wedge$ (and), $\vee$ (or), $\Rightarrow$ (implication), $\Leftrightarrow$ (biconditional); definition of a well-formed formula (wff); representation of simple word problems as wff (this can be used for motivation); the values **true** and **false**; interpretation of a wff; truth tables; satisfiable, unsatisfiable and valid formulae.

Equivalence laws: commutativity of $\wedge$, $\vee$; associativity of $\wedge$, $\vee$; distributivity; de Morgan's laws; law of implication ($p \Rightarrow q \equiv \sim p \vee q$); law of biconditional ($(p \Leftrightarrow q) \equiv (p \Rightarrow q) \wedge (q \Rightarrow p)$); identity ($p \equiv p$); law of negation ($\sim(\sim p) \equiv p$); law of excluded middle ($p \vee \sim p \equiv$ **true**); law of contradiction($p \wedge \sim p \equiv$ **false**); simplification rules for $\wedge$, $\vee$.

$p \vee p \equiv p$	$p \wedge p \equiv p$
$p \vee \textbf{true} \equiv \textbf{true}$	$p \wedge \textbf{true} \equiv p$
$p \vee \textbf{false} \equiv p$	$p \wedge \textbf{false} \equiv \textbf{false}$
$p \vee (p \wedge q) \equiv p$	$p \wedge (p \vee q) \equiv p$

The equivalence rules can be used to simplify propositional wffs, for example:

1) $(p \Rightarrow q) \wedge (p \Rightarrow r)$ to $p \Rightarrow (q \wedge r)$

2) $((p \Rightarrow q) \wedge p) \Rightarrow q$ to **true**

etc.

) Binary valued quantities; basic postulates of Boolean algebra; operations AND, OR and NOT; truth tables.

(c) Basic theorems of Boolean algebra (e.g. Duality, idempotence, commutativity, associativity, distributivity, operations with 0 and 1, complements, absorption, involution); De Morgan's theorem and its applications; reducing Boolean expressions to sum of products and product of sums forms; Karnaugh maps (up to four variables).

Verify the laws of boolean algebra using truth tables. Inputs, outputs for circuits like half and full adders, majority circuit etc., SOP representation; reduction using Karnaugh maps and boolean algebra.

Computer Hardware

(a) Elementary logic gates (NOT, AND, OR, NAND, NOR, XOR, XNOR) and their use in circuits.

(b) Applications of Boolean algebra and logic gates to half adders, full adders, encoders, decoders, multiplexers, NAND, NOR as universal gates.

Show the correspondence between boolean functions and the corresponding switching circuits or gates. Show that NAND and NOR gates are universal by converting some circuits to purely NAND or NOR gates.

SECTION B

The programming element in the syllabus (Sections B nd C) is aimed at algorithmic problem solving and ***not*** *merely rote learning of Java syntax. The Java ersion used should be 1.5 or later. For programming, he students can use any text editor and the javac and java programs or any development environment: for xample, BlueJ, Eclipse, NetBeans etc. BlueJ is trongly recommended for its simplicity, ease of use and because it is very well suited for an 'objects first' pproach.*

3. ***Programming in Java (Review of Class XI Sections B and C)***

Note that items 4 to 8 will get introduced almost simultaneously when classes and their definitions are introduced.

4. ***Objects***

(a) Objects as data (attributes) + behaviour (methods or functions); object as an instance of a class. Constructors.

Difference between object and class should be made very clear. BlueJ (www.bluej.org) and Greenfoot (www.greenfoot.org) can be profitably used for this purpose. Constructor as a special kind of function; the *new* operator; multiple constructors with different argument structures; constructor returns a reference to the object.

(b) Analysis of some real world programming examples in terms of objects and classes.

Use simple examples like a calculator, date, number, etc. to illustrate how they can be treated as objects that behave in certain well-defined ways and how the interface provides a way to access behaviour. Illustrate behaviour changes by adding new functions, deleting old functions or modifying existing functions.

5. ***Primitive values, wrapper classes, types and casting***

Primitive values and types: int, short, long, float, double, boolean, char. Corresponding wrapper classes for each primitive type. Class as type of the object. Class as mechanism for user defined types. Changing types through user defined casting and automatic type coercion for some primitive types.

Ideally, everything should be a class; primitive types are defined for efficiency reasons; each primitive type has a corresponding wrapper class. Classes as user defined types. In some cases types are changed by automatic coercion or casting – e.g. mixed type expressions. However, casting in general is not a good idea and should be avoided, if possible.

Variables, expressions

Variables as names for values; expressions (arithmetic and logical) and their evaluation (operators, associativity, precedence). Assignment operation; difference between left hand side and right hand side of assignment.

Variables denote values; variables are already defined as attributes in classes; variables have types that constrain the values it can denote. Difference between variables denoting primitive values and object values – variables denoting objects are references to those objects. The assignment operator = is special. The variable on the lhs of = denotes the memory location while the same variable on the rhs denotes the contents of the location e.g. i=i+2.

7. Statements, scope

Statements; conditional (if, if-then-else, switch-break, ?: ternary operator), looping (for, while-do, do-while, continue, break); grouping statements in blocks, scope and visibility of variables.

Describe the semantics of the conditional and looping statements in detail. Evaluation of the condition in conditional statements (esp. difference between || and | and && and &). Emphasize fall through in *switch* statement. Many small examples should be done to illustrate control structures. Printing different kinds of patterns for looping is instructive. When number of iterations are known in advance use the *for* loop otherwise the *while-do* or *do-while* loop. Express one loop construct using the others. For e.g.:

for (<init>; <test>; <inc>) <stmt>; is equivalent to:

Using while

<init>; while <test> {<stmt>; <inc> }

Using do-while

<init>; if !<test> do <stmt>; <inc> while <test>;

Nesting of blocks. Variables with block scope, function scope, class scope. Visibility rules when variables with the same name are defined in different scopes.

8. Functions

Functions/methods (as abstractions for complex user defined operations on objects), functions as mechanisms for side effects; formal arguments and actual arguments in functions; different behaviour of primitive and object arguments. Static functions and variables. The this variable. Examples of algorithmic problem solving using functions (various number theoretic problems, finding roots of algebraic equations).

Functions are like complex operations where the object is implicitly the first argument. Variable **this** denotes the current object. Functions typically return values, they may also cause side-effects (e.g. change attribute values of objects) – typically functions that are only supposed to cause side-effects return *void* (e.g. *Set* functions). Java passes argument by value. Illustrate the difference between primitive values and object values as arguments (changes made inside functions persist after the call for object values). Static definitions as class variables and class functions visible and shared by all instances. Need for static functions and variables. Introduce the *main* method – needed to begin execution.

9. Arrays, strings

(a) *Structured data types – arrays (single and multi-dimensional), strings. Example algorithms that use structured data types (e.g. searching, finding maximum/minimum, sorting, solving systems of linear equations, substring, concatenation, length, access to char in string, etc.).*

Storing many data elements of the same type requires structured data types – like arrays. Access in arrays is constant time and does not depend on the number of elements. Structured data types can be defined by classes – *String*. Introduce the Java library *String* class and the basic operations on strings (accessing individual characters, various substring operations, concatenation, replacement, index of operations). The Class *StringBuffer* should be introduced for those applications that involve heavy manipulation of strings.

Basic concept of a virtual machine; Java virtual machine; compilation and execution of Java programs (the javac and java programs).

The JVM is a machine but built as a program and not through hardware. Therefore it is called a virtual machine. To run, JVM machine language programs require an interpreter (the *java* program). The advantage is that such JVM machine language programs (.class files) are portable and can run on any machine that has the *java* program.

(c) *Compile time and run time errors; basic concept of an exception, the Exception class, catch and throw.*

Differentiate between compile time and run time errors. Run time errors crash the program. Recovery is possible by the use of exceptions. Explain how an exception object is created and passed up until a matching *catch* is found. This behaviour is different from the one where a value is returned by a deeply nested function call. It is enough to discuss the *Exception* class. Sub-classes of *Exception* can be discussed after inheritance has been done in Class XII.

(d) *Class as a contract; separating implementation from interface; encapsulation; private and public.*

Class is the basic reusable unit. Its function prototypes (i.e. the interface) work as a visible contract with the outside world since others will use these functions in their programs. This leads to encapsulation (i.e. hiding implementation information) which in turn leads to the use of private and public for realizing encapsulation.

(e) *Interfaces in Java; implementing interfaces through a class; interfaces for user defined implementation of behaviour.*

Motivation for interface: often when creating reusable classes, some parts of the exact implementation can only be provided by the final end user. For example, in a class that sorts records of different types the exact comparison operation can only be provided by the end user. Since only he/she knows which field(s) will be used for doing the comparison and whether sorting should be in ascending or descending order be given by the user of the class.

Emphasize the difference between the Java language construct *interface* and the word interface often used to describe the set of function prototypes of a class.

(f) *Basic input/output using Scanner and Printer classes from JDK; files and their representation using the File class, file input/output; input/output exceptions. Tokens in an input stream, concept of whitespace, extracting tokens from an input stream (StringTokenizer class).*

The *Scanner* class can be used for input of various types of data (e.g. int, float, char etc.) from the standard input stream or a file input stream. The *File* class is used model file objects in the underlying system in an OS independent manner. Similarly, the *Printer* class handles output. Only basic input and output using these classes should be covered.

Discuss the concept of a token (a delimited continuous stream of characters that is meaningful in the application program – e.g. words in a sentence where the delimiter is the blank character). This naturally leads to the idea of delimiters and in particular whitespace and user defined characters as delimiters. As an example show how the *StringTokenizer* class allows one to extract a sequence of tokens from a string with user defined delimiters.

(g) *Concept of recursion, simple recursive functions (e.g. factorial, GCD, binary search, conversion of representations of numbers between different bases).*

Many problems can be solved very elegantly by observing that the solution can be composed of solutions to 'smaller' versions of the same problem with the base version having a known simple solution. Recursion can be initially motivated by using recursive equations to define certain functions. These definitions are fairly obvious and are easy to understand. The definitions can be directly converted to a program. Emphasize that any recursion must have a base case. Otherwise, the computation can go into an infinite loop. Illustrate this by removing the base case and running the program. Examples:

(i) Definition of factorial:

factorial(0) = 1 //base case

*factorial(n) = n * factorial(n-1)*

(ii) Definition of GCD:

gcd(m, n) =

if (m==n) then n //base case

else if (m>n) then gcd(m-n, n)

else gcd(m, n-m)

(iii) Definition of Fibonacci numbers:

fib(0) = 1 //base case

fib(1) = 1 //base case

fib(n) = fib(n-1)+ fib(n-2)

The tower of Hanoi is a very good example of how recursion gives a very simple and elegant solution where as non-recursive solutions are quite complex. Discuss the use of a stack to keep track of function calls. A stack can also be used to solve the tower of Hanoi problem non-recursively.

SECTION C

Inheritance, polymorphism, data structures, computational complexity

10. *Inheritance and polymorphism*

Inheritance; base and derived classes; member access in derived classes; redefinition of variables and functions in subclasses; abstract classes; class Object; protected visibility. Subclass polymorphism and dynamic binding.

Emphasize the following:

- inheritance as a mechanism to reuse a class by extending it.
- inheritance should not normally be used just to reuse some functions defined in a class but only when there is a genuine specialization (or subclass) relationship between objects of the base class and that of the derived class.
- Allows one to implement operations at the highest relevant level of abstraction.
- Freezes the interface in the form of abstract classes with abstract functions that can be extended by the concrete implementing classes. For example, an abstract class *Shape* can have an abstract function draw that is implemented differently in the sub-classes like *Circle*, *Quadrilateral* etc.
- how the exact function call at run time depends on the type of the object referenced by the variable. This gives sub-class polymorphism. For example in the code fragment:

  Shape s1=new Circle(), s2=new Quadrilateral();

  s1.draw(); //the draw is the draw in Circle

  s2.draw(); //the draw is the draw in Quadrilateral

  the two *draw* function invocations on *s1*, *s2* invoke different *draw* functions depending on the type of objects referenced by *s1* and *s2* respectively.

11. *Data structures*

(a) *Basic data structures (stack, queue, dequeue); implementation directly through classes; definition through an interface and multiple implementations by implementing the interface. Basic algorithms using the above data structures.*

A data structure is a data collection with well defined operations and behaviour or properties. The behaviour or properties can usually be expressed formally using equations or some kind of logical formulae. Consider for e.g. a stack with operations defined as follows:

void push(Object o)

Object pop()

boolean isEmpty()

Object top()

Then, for example the LIFO property can be expressed by (assume s is a stack):

if *s.push(o); o1=pop()* then *o* ≡ *o1*

What the rule says is: if *o* is pushed on the stack *s* and then it is popped and *o1* is the object obtained then *o, o1* are identical.

Another useful property is:

if *s.isEmpty() == true* then *s.pop() = ERROR*

It says that popping an empty stack gives ERROR.

Similarly, several other properties can also be specified. It is important to emphasize the behavioural rules or properties of a data structure since any implementation must guarantee that the rules hold.

Some simple algorithms that use the data structures:

(i) For stack: parentheses matching, tower of Hanoi, nested function calls; solving a maze.

(ii) For queue: scheduling processes, printers, jobs in a machine shop.

(b) Recursive data structures: singly and doubly linked lists, binary trees, tree traversals, binary search tree. Algorithms using these data structures (merge sort and quick sort, searching).

Data structures should be defined as abstract data types with a well defined interface (it is instructive to define them using the Java interface construct) – see the comments in (a) above. Emphasize that algorithms for recursive data structures are themselves recursive and that algorithms are usually the simplest and most elegant. The following should be covered for each data structure:

Lists: insertion, deletion, reversal, appending two lists, extracting an element or a sublist, checking emptiness. Searching, sorting (by quicksort and mergesort algorithms), binary search in a sorted list.

Binary trees: apart from the definition the following concepts should be covered: external and internal nodes, height, completeness, balancing, Traversals (pre, post and in-order). Implementation using arrays and linked structures.

Binary search tree: insertion, deletion, search.

12. Complexity and big O notation

Concrete computational complexity; concept of input size; estimating complexity in terms of functions; importance of dominant term; best, average and worst case. Big O notation for computational complexity; analysis of complexity of example algorithms using the big O notation (e.g. Various searching and sorting algorithms, algorithm for solution of linear equations etc.).

Points to be given particular emphasis:

(i) Algorithms are usually compared along two dimensions – amount of space (that is memory) used and the time taken. Of the two the time taken is usually considered the more important. The motivation to study time complexity is to compare different algorithms and use the one that is the most efficient in a particular situation.

(ii) Actual run time on a particular computer is not a good basis for comparison since it depends heavily on the speed of the computer, the total amount of RAM in the computer, the OS running on the system and the quality of the compiler used. So we need a more abstract way to compare the time complexity of algorithms.

(iii) This is done by trying to approximate the number of operations done by each algorithm as a function of the size of the input. In most programs the loops are important in deciding the complexity. For example in bubble sort there are two nested loops and in the worst case the time taken will be proportional to n(n-1) where n is the number of elements to be sorted. Similarly, in linear search in the worst case the target has to be compared with all the elements so time taken will be proportional to n where n is the number of elements in the search set.

(iv) In most algorithms the actual complexity for a particular input can vary. For example in search the number of comparisons can vary from 1 to n. This means we need to study the best, worst and average cases. Comparisons are usually made taking the worst case. Average cases are harder to estimate since it depends on how the data is distributed. For example in search, if the elements are uniformly distributed it will take on the average n/2 comparisons when the average is taken over a statistically significant number of instances.

(v) Comparisons are normally made for large values of the input size. This means that the dominant term in the function is the important term. For example if we are looking at bubble sort and see that time taken can be estimated as: $a*n^2 + b*n + c$ where n is the number of elements to be sorted and a, b, c are constants then for large n the dominant term is clearly n^2 and we can, in effect, ignore the other two terms.

All the above motivates the big O notation. Let $f(n)$, $g(n)$ be positive functions, then $f(n)$ is said to be $O(g(n))$ if there exists constants c, n_0 such that $f(x) \leq c*g(n)$ whenever $n>n_0$. What this means is that $g(n)$ asymptotically dominates $f(n)$. Expressing time complexity using the big O notation gives us an abstract basis for comparison and frees us from bothering about constants. So the estimated time complexity $a*n^2+b*n+c$ is $O(n^2)$.

Analyse the big O complexity of the algorithms pertaining to the data structures in 11 (a) and (b) above.

13. *Implementation of algorithms to solve problems*

The students are required to do lab assignments in the computer lab concurrently with the lectures. Programming assignments should be done such that each major topic is covered in at least one assignment. Assignment problems should be designed so that they are non-trivial and make the student do algorithm design, address correctness issues, implement and execute the algorithm in Java and debug where necessary.

Self explanatory.

PAPER II - PRACTICAL

is paper of three hours duration will be evaluated the Visiting Examiner appointed locally and approved by the Council.

e paper shall consist of three programming problems from which a candidate has to attempt any one. The practical consists of the two parts:

Planning Session

2. Examination Session

The total time to be spent on the Planning session and the Examination session is three hours. After completing the Planning session the candidates may begin with the Examination session. A maximum of 90 minutes is permitted for the Planning session. However, if the candidates finish earlier, they are to be permitted to begin with the Examination session.

Planning Session

The candidates will be required to prepare an algorithm and a hand written Java program to solve the problem.

Examination Session

The program handed in at the end of the Planning session shall be returned to the candidates. The candidates will be required to key-in and execute the Java program on seen and unseen inputs individually on the Computer and show execution to the Visiting Examiner. A printout of the program listing including output results should be attached to the answer script containing the algorithm and handwritten program. This should be returned to the examiner. The program should be sufficiently documented so that the algorithm, representation and development process is clear from reading the program. Large differences between the planned program and the printout will result in loss of marks.

Teachers should maintain a record of all the assignments done as part of the practical work through the year and give it due credit at the time of cumulative evaluation at the end of the year. Students are expected to do a ***minimum*** *of twenty assignments for the year.*

Marks (out of a total of 100) should be distributed as given below:

Continuous Evaluation

Candidates will be required to submit a work file containing the practical work related to programming assignments done during the year.

Programming assignments done throughout the year (Internal evaluation) - 10 marks

Programming assignments done throughout the year (Visiting Examiner) - 10 marks

erminal Evaluation

Solution to programming problem on the computer - 60 marks

(Marks should be given for choice of algorithm and implementation strategy, documentation, correct output on known inputs mentioned in the question paper, correct output for unknown inputs available only to the examiner.)

Viva-voce - 20 marks

(Viva-voce includes questions on the following aspects of the problem attempted by the student: the algorithm and implementation strategy, documentation, correctness, alternative algorithms or implementations. Questions should be confined largely to the problem the student has attempted).

NOTE:

Algorithm should be expressed clearly using any standard scheme such as a pseudo code.

EQUIPMENT

There should be enough computers to provide for a teaching schedule where at least three-fourths of the time available is used for programming.

Schools should have equipment/platforms such that all the software required for practical work runs properly, i.e. it should run at acceptable speeds.

Since hardware and software evolve and change very rapidly, the schools may have to upgrade them as required. Following are the recommended specifications as of now:

The Facilities:

- *A lecture cum demonstration room with a MULTIMEDIA PROJECTOR/ an LCD and O.H.P. attached to the computer.*
- *A white board with white board markers should be available.*
- *A fully equipped Computer Laboratory that allows one computer per student.*
- *Internet connection for accessing the World Wide Web and email facility.*
- *The computers should have a minimum of 256 MB (512MB preferred) RAM and a PIII or higher processor. The basic requirement is that it should run the operating system and Java programming system (Java compiler, Java runtime environment, Java development environment) at acceptable speeds.*
- *Good Quality printers.*

Software:

- *Any suitable Operating System can be used.*
- *JDK 1.5 or later.*
- *Documentation for the JDK version being used.*
- *A suitable text editor. A development environment with a debugger is preferred (e.g. BlueJ, Eclipse, NetBeans). BlueJ is recommended for its ease of use and simplicity.*